For Macintosh Programmers & Developers

In This Issue!
Intro To
OpenDoc
Development
Framework

MacTech™
AGAZINE

. 11, No. 11 • November 1995

NSIDE:

ETTING STARTED
CL and Visual Architect, Part 2

ROGRAMMER'S CHALLENGE
nclosing Bounds

YMANTEC TOP 10

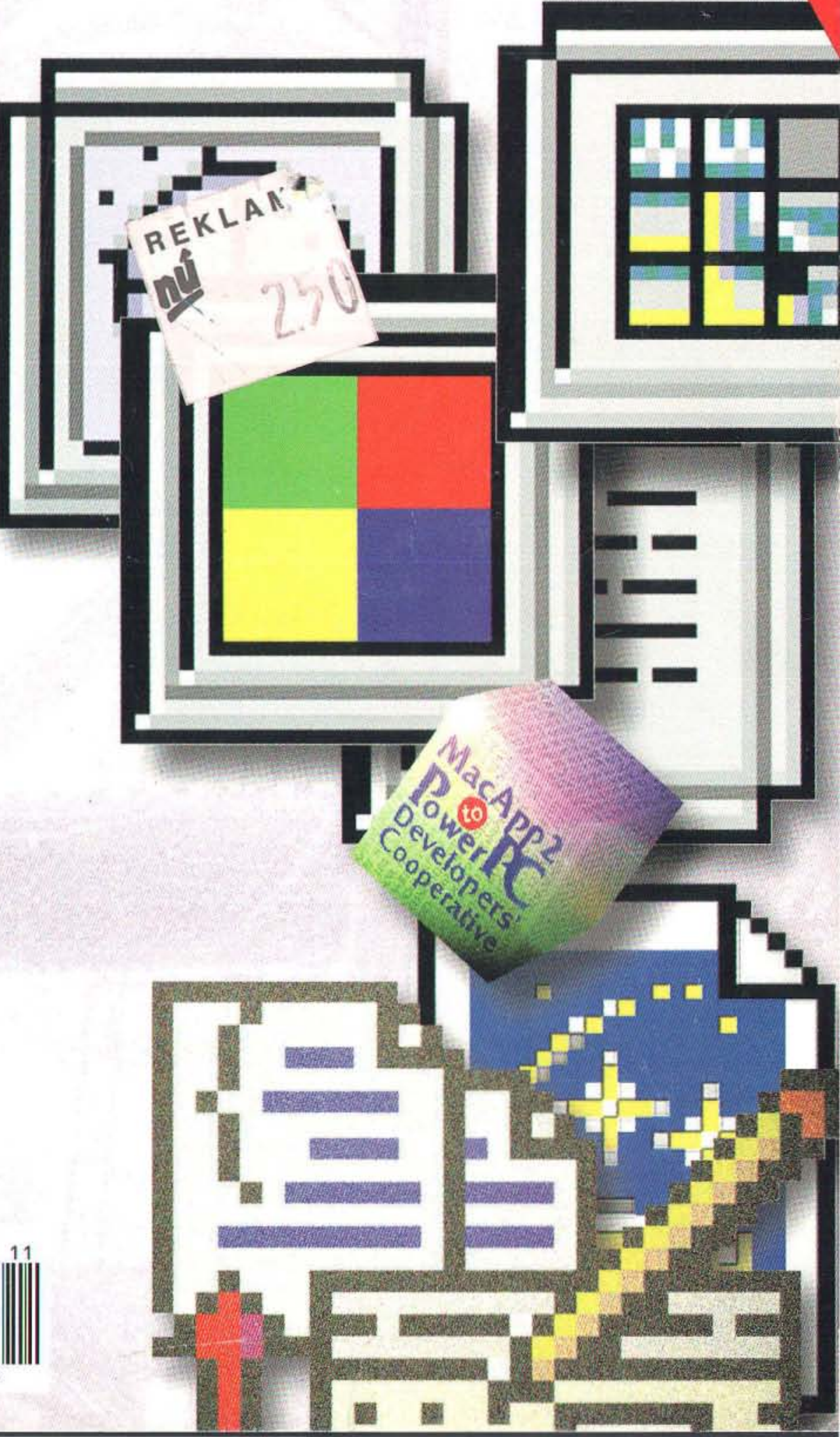
ASCAL ACTIVISTS
lacApp Pascal Rides Again

RABB'S APPLE
isually Speaking

PPLE TECHNOLOGY
he OpenDoc Development Framework

NIFORM RESOURCE LOCATORS

ND MORE!



REKLAM
nu
250

$5.85 US
$6.95 Canada
ISSN 1067-8360
Printed in U.S.A.

11
0 32128 74887 8

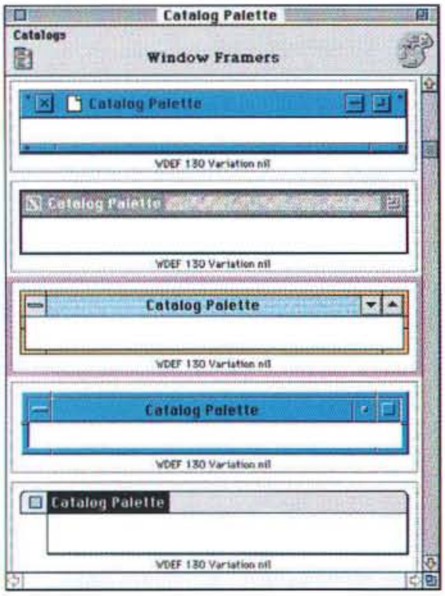
Building Internet And Other Native PowerPC Applications Has Never Been Easier Or Faster.

SmalltalkAgents®

Agents Object System
MACINTOSH

SmalltalkAgents (STA) is a sophisticated rapid application development environment based on a new generation of the Smalltalk language, enabling you to easily deliver double-clickable applications.

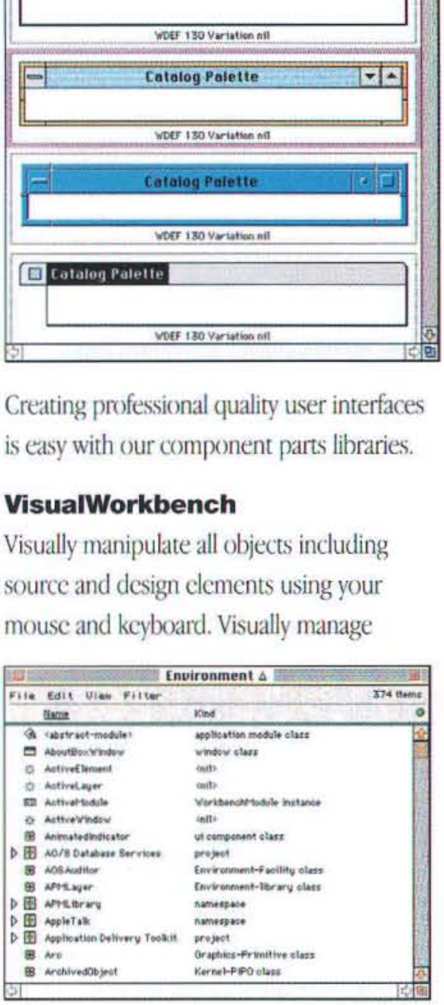
Copland Style GUI Look & Feel



Creating professional quality user interfaces is easy with our component parts libraries.

VisualWorkbench

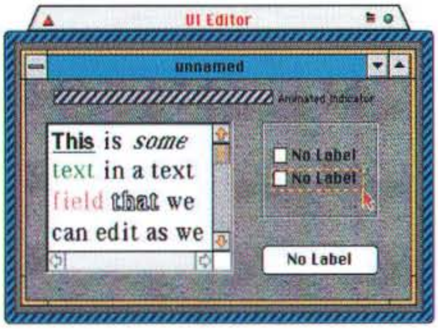
Visually manipulate all objects including source and design elements using your mouse and keyboard. Visually manage design and project elements in a "Finder-like" desktop workspace as fluidly as you work with folders and documents on your desktop. Interactively build, wire, and interconnect reusable components and interfaces in an integrated environment.



Name	Kind
<abstract-module>	application module class
AboutBoxWindow	window class
ActiveElement	<nil>
ActiveLayer	<nil>
ActiveModule	WorkbenchModule instance
ActiveWindow	<nil>
AnimatedIndicator	ui component class
AO/S Database Services	project
AOSAuditor	Environment-Facility class
APMLayer	Environment-library class
APMLibrary	namespace
AppleTalk	namespace
Application Delivery Toolkit	project
Arc	Graphics-Primitive class
ArchivedObject	Kernel-PIPO class

GUI Design & Generation

Live "Drag and Drop" manipulation to build your application's visual interface using components that "know" how to behave and autoconfigure themselves into an environment. Create new components and/or wire together existing components that can be saved as reusable template designs for use in other applications or containers.



This is *some* text in a text field that we can edit as we

DTP Engine & Word Processor

STA not only includes a programmable word processor component and HyperMedia engine, but also a powerful report writer that supports embedding of any kind of objects, movies, flows, and international text, and page layout.

C/C++, Pascal Workbench

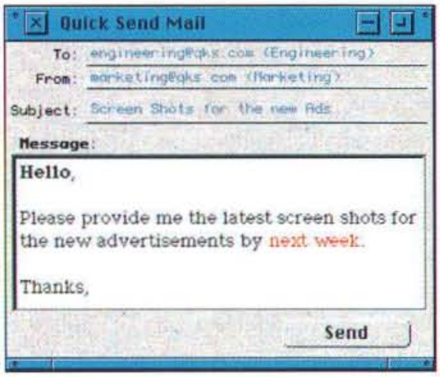
Compile, edit, and dynamically link C/C++, Pascal, Fortran, and Assembly code directly from within our STA VisualWorkbench as an integrated part of the Smalltalk application development process.

Component-based Architecture

STA components are designed for OpenDoc and OLE, and will give you transparent integration with OpenDoc and OLE when they become available.

Threading & Internet Tools

STA provides powerful support for Internet server as well as client tool development. Pre-emptive threading, thread safe libraries and classes for TCP/IP protocols are standard features enabling you to quickly and easily deliver custom e-mail, WWW, list-server, and other dial-up/network related apps.



To: engineering@qks.com (Engineering)
From: marketing@qks.com (Marketing)
Subject: Screen Shots for the new Ads
Message:

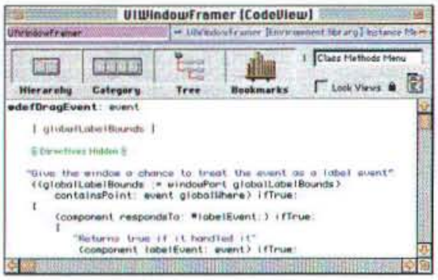
Hello,

Please provide me the latest screen shots for the new advertisements by next week.

Thanks,

PowerPC Support

STA provides binary portability across different CPUs and Operating Systems. Design applications today on one platform and simply deploy on other platforms as required.


UIWindowFramer [CodeView]
UIWindowFramer
UIWindowFramer [Environment-library] Instance Me
Hierarchy
Category
Tree
Bookmarks
Class Methods Menu
Look Views

```
wdefDragEvent: event

    | globalLabelBounds |

   § Directives Hidden §

  "Give the window a chance to treat the event as a label event"
   ((globalLabelBounds := windowPort globalLabelBounds)
       containsPoint: event globalWhere) ifTrue:
   [
       (component respondsTo: #labelEvent:) ifTrue:
       [
          "Returns true if it handled it"
           (component labelEvent: event) ifTrue:
```

Contact us about our Web Server & Client Toolkit at 1-800-296-1339 or at <info@qks.com> or visit our Web site http://www.qks.com/.

Quasar Knowledge Systems, Inc.
9818 Parkwood Drive
Bethesda, MD 20814 USA
Tel:(301)530-4853 Fax:(301)530-5712

© Quasar Knowledge Systems, Inc. 1995 All rights reserved. All other brand or product names are trademarks or registered trademarks of their respective holders.

Eddy Award Winner** for **Best New Developer Tool
– MacUser Editors Choice Awards, 1993

"A distinct improvement over ResEdit."
– MacTech / MacTutor

"Resorcerer's data template system is amazing!"
– Bill Goodman, author of Compact Pro

"Nuke ResEdit! Resorcerer is mission-critical for us."
– Dave Winer, Userland Frontier

"The color pixel editors are wonderful! A work of art!"
– Dave Winzler, author of Microseeds Redux

"Every Macintosh developer should own a copy of Resorcerer."
– Leonard Rosenthol, Aladdin Systems

"Resorcerer will pay for itself many times over in saved time and effort."
– MacUser review

"The template that disassembles 'PICT's is awesome!"
– Bill Steinberg, author of Pyro! and PBTools

"Resorcerer proved indispensible in its own creation!"
– Doug McKenna, author of Resorcerer

"...a wealth of time-saving tools."
MacUser Review, Dec. 1992

MACUSER EDITORS' CHOICE AWARD
1993
WINNER

RESORCERER®

Version 1.2.4

The Resource Editor for the Macintosh Wizard

ORDERING INFO

Needs: ≥Mac Plus, ≥ Sys 4.2, 1MB
Likes: ≥Mac Plus, ≥ Sys 7.0, 2MB
32-bit clean, AU/X compatible

Price: $256 (decimal)
(Educational, quantity, or
other discounts available)

Includes: 500 page manual
60-day Money-Back Guarantee
Domestic UPS ground shipping

Payment: Check, PO's, or Visa/MC

Extras (call us):
COD, FedEx, UPS Blue/Red,
International Shipping

Downloadable Demos/Updaters:
AppleLink: Software Sampler
AOL: Software Libs/Development
CompuServe: MACDEV/Tools
or call us.

New 1.2 Features:

- **New 'cicn', 'ppat', 'crsr', 'acur', 'pltt', 'clut' editors**
- **Powerful icon family editing (all 9 icon types)**
- **Color pixel anti-aliasing, dithering, and lots more**
- **Complete 'PICT' disassembly and reassembly**
- **Resource sorting; ROM resource browsing**
- **120 template field parsing types now supported**
- **New insertion & deletion template field types**
- **Text-only 'PICT' resources**
- **Lots of improvements throughout**

- Easier, faster, more Mac-like, and more productive than ResEdit
- Safer memory-based, not disk-file-based, design and operation
- All file information and common commands in one easy-to-use window
- Compares resource files, and even **edits your data forks** as well
- Visible, accumulating, editable scrap
- Searches and opens/marks/selects resources by text content
- Makes global resource ID or type changes easily and safely
- Builds resource files from simple Rez-like scripts
- Most editors DeRez directly to the clipboard
- All graphic editors support screen-copying or partial screen-copying
- Hot-linking Value Converter for editing 32 bits in a dozen formats
- Its own 32-bit List Mgr can open and edit very large data structures
- Templates can pre- and post-process any arbitrary data structure
- Includes nearly 200 templates for common system resources
- TMPLs for Installer, MacApp, QT, Help, AppleEvent, OCE, GX, etc.
- Full integrated support for editing color dialogs and menus
- Try out balloons, 'ictb's, lists and popups, even create C source code
- Integrated single-window Hex/Code Editor, with patching, searching
- Editors for cursors, versions, pictures, bundles, and lots more
- Well-designed, helpful developer tools being added all the time
- Relied on by thousands of Macintosh developers around the world

MATHEMÆSTHETICS, INC.
P.O. Box 298 • Boulder • CO • 80306-0298 • USA
Phone: (303) 440-0707 • Fax: (303) 440-0504
AppleLink/AmericaOnline: RESORCERER • Internet: resorcerer@aol.com

For Macintosh Programmers & Developers
MacTech™ MAGAZINE
A publication of XPLAIN CORPORATION

How To Communicate With Us

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail?** If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

DEPARTMENTS	Internet	eWorld	CompuServe	AppleLink	America Online
Orders, Circulation, & Customer Service	custservice@mactech.com	MT.CustSvc	71333,1063	MT.CUSTSVC	MT CUSTSVC
Editorial	editorial@mactech.com	MT.Editors	71333,1065	MT.EDITORIAL	MT EDITORS
Programmer's Challenge	progchallenge@mactech.com	MT.PrgChal	71552,174	MT.PROGCHAL	MT PRGCHAL
Ad Sales	adsales@mactech.com	MT.AdSales	71552,172	MT.ADSALES	MT ADSALES
Online Support	online@mactech.com	MT.Online	–	–	–
Accounting	accounting@mactech.com	accounting@xplain.com	–	–	
Marketing	marketing@mactech.com	MT.Mktg	–	–	–
Press Releases	pressreleases@mactech.com	pressreleases@xplain.com	–	–	–
General	info@mactech.com	info@xplain.com	71333,1064	MACTECHMAG	MacTechMag
Online support area	http://www.mactech.com	*use shortcut:* MACTECH	*type* GO MACTECHMAG	*see* Third Parties: Third Parties (H-O)	*use keyword:* MACTECHMAG

Xplain Corporation

Chief Executive Officer • Neil Ticktin
Chief Financial Officer • Andrea J. Sniderman
Advertising Executive • Ruth Subrin
Customer Service • Al Estrada
Art Director • Judith Chaplin
Software Engineer • Don Bresee
Accounting Assistant • Brian Shin
Network Administrator • Donal Corcoran
Administrative Assistant • Susan Pomrantz

Board of Advisors • Blake Park, Alan Carsrud, Jordan Mattson, Steven Geller

Printed on recycled paper.

PRINTED WITH SOY INK

MacTech Magazine

*MacTech Magazine is grateful to the following individuals who contribute on a regular basis. We encourage others to share the technology. We are dedicated to the distribution of useful programming information without regard to Apple's developer status. For information on submitting articles, ask us for our **writer's kit** which includes the terms and conditions upon which we publish articles.*

Regular Columnists

Getting Started • Dave Mark
Programmer's Challenge • Bob Boonstra
Inside Info • Chris Espinosa
Symantec Top 10 • Symantec Technical Support
Crabb's Apple • Don Crabb
URLs • Jim Straus

Contributing Editors

Copland • Steve Kiene, Mindvision
Database • Liam Breck
Internet • Jon Wiederspan
MagicCap/Telescript • Richard Clark, General Magic
Misc. Topics • Eric Gundrum
Performance Programming • Jim Gochee
Tips & Tidbits (and Technical Editor) • Steve Sisak

And our Editors...

Publisher/Editor-in-Chief • Neil Ticktin
Managing Editor • Matt Neuburg
Editorial Assistant • John Kawakami
Editor-at-Large • Scott T Boyd

The names MacTech, MacTech Magazine, MacTutor and the MacTutorMan logo are trademarks and registered trademarks of Xplain Corporation. All contents are copyright 1984-1995 by Xplain Corporation. All rights reserved. Trademarks appearing in MacTech Magazine remain the property of the companies that hold license.

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are $47.00 per year. Canadian subscriptions are $59.00 per year. All other international subscriptions are $97.00 per year. Domestic source code disk subscriptions are $77 per year. All international disk subscriptions are $97.00 a year. Please remit in U.S. funds only. Second Class postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

TABLE OF CONTENTS

For Macintosh Programmers & Developers

MacTech™ MAGAZINE

November 1995 • Volume 11, No. 11

By Scott T Boyd, Editor-at-Large
http://www.hax.com/

Well, as of this writing we still haven't heard anything of substance from Apple about the MacHack Top Ten <http://www.machack.com/>. Sure, they released a statement about the top 5 items, but it was the same old "important future direction" verbiage we so often hear from Apple. From Apple's initial response:

> "Stay tuned...as we work out plans..."
> "There's a lot of support for this as an idea..."
> "We are investigating what would be involved..."
> "We'll consider validating it..."
> "We're evaluating..."
> "Any volunteers?"
> "It's a goal..."
> "A topic of hot discussion..."
> "...as that happens we'll have a better vehicle..."
> "Keep tracking our product announcements..."

[Thanks to Al Evans <al@powertools.com> for picking out the operative phrases.]

Funny, I didn't think much of the whole Top 10 Developer Issues list idea at the conference, but Apple has committed to address the issues, so I find their lack of action disappointing.

It simply is *not* common practice for Apple to do market research on the needs and wants of Macintosh developers. Therefore, it shouldn't be much of a surprise that Apple's relationship with developers lacks a few items.

It's time for Apple to start delivering solutions to other problems that have long plagued developers. Let's talk about a few constructive ideas.

Information

At twenty seven volumes and growing, Inside Macintosh is starting to represent a significant investment for developers, in terms of both time and money. Even so, developers could use much more. For example, as Sean Parent, Photoshop engineer and former Blue Meanie, recently said:

> "Apple, what I want is a clear and precise explanation of every external public interface. I want to know what all the limitations, side effects, bugs, exceptional behavior, invariants, assumptions, version history, and supported parameter ranges are. Inside Mac gives me about 10% of that."

Once we have a good start on that, publishing the information on the Internet and on cheap CD's would make a great next step.

Overcoming limitations

TextEdit, the Menu Manager, the Dialog Manager, and the List Manager have long represented Apple's strange affinity for living with arbitrary limitations. They have all evolved just about as far as they can. Maybe it's time to offer up some new, improved toolbox managers. While it runs the risk of dying of not-invented-here syndrome, would it be that far-fetched for Apple to license some third-party solutions?

For example, imagine the benefit to developers should Apple decide to do away with the 32K TextEdit limit by taking Chris Thomas' recent suggestion on Semper.Fi, "Just put WASTE in a shared library, license it for inclusion with System 7.6 and beyond, and be done with it." Well, write some docs and test it a bit, too.

Bringing Apple applications into the 90's

Where do we start on this one? OpenDoc support in the Finder and MPW? AppleScript support for Apple control panels, SimpleText, MPW, and others?

Attention

Developers need Apple's attention. Apple needs developers' attention. It all comes down to rebuilding the relationship between Apple and developers, but we'll save some of this for later. Don't forget to check out the web site for further developments, which Apple has promised by the beginning of October.

PICTURE, IF YOU WILL...

Rarely does Apple put out advertising that we can all be proud of. Sure, there have been ads that made us feel good, but deep down we usually know that the ads are preaching to the choir. When was the last time we saw an advertising campaign that we all just *knew* sold oodles of machines? I don't even know if the "great" 1984 commercial sold many machines.

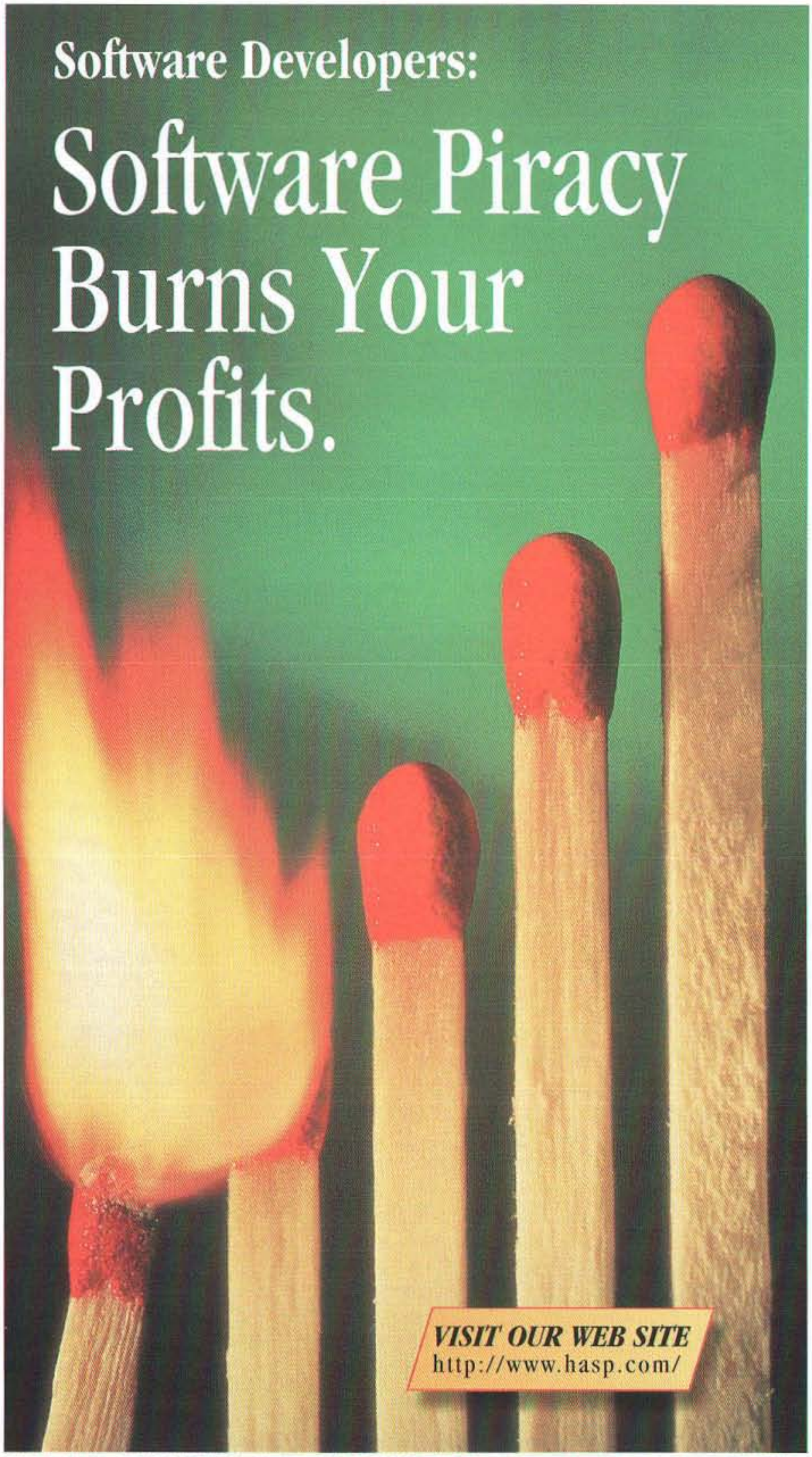
To be fair, Apple used to have some good ads which managed to deliver the "wow!" factor. I recently thought that they'd made a comeback, but I was rudely surprised to find out that they were actually Compaq ads. I was shocked.

However, it's all the rage to drive $10B companies from the back seat lately, so here goes. If I was in charge, I'd probably try a few wacky things. First ad campaign? Show developers writing software on the Macintosh. We don't need nuns with pagers or corporate execs with surfboards, we've got real programmers putting together major-league products using Macintosh.

Continued on page 68

Software Developers:

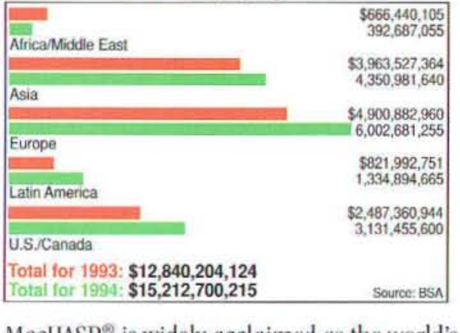
Software Piracy Burns Your Profits.



VISIT OUR WEB SITE
http://www.hasp.com/

Each year, the illegal use of software consumes nearly 50% of your potential revenues. With the flames of piracy eating away at your profits, can you afford not to protect your software?

Software Obtained Illegally, by region, 1993 vs. 1994



Source: BSA

MacHASP® is widely acclaimed as the world's most advanced software protection solution for Macintosh computers. Since 1984, thousands of leading Mac and PC developers have used over one million MacHASP and HASP keys to protect billions of dollars worth of software. Why? Because MacHASP's security, reliability, and ease-of-use led them to a simple conclusion: MacHASP is the most effective software protection system available.

Today, more developers are choosing MacHASP than any other software protection method. To learn why, and to see how easily you can increase your revenues, call now to order your MacHASP Developer's Kit.

1-800-223-4277

ALADDIN

The Professional's Choice

North America — **Aladdin Software Security Inc.**
Tel: (800) 223 4277, 212-564 5678
Fax: 212-564 3377
E-mail: sales@hasp.com
WWW: http://www.hasp.com/

Intl Office — **Aladdin Knowledge Systems Ltd.**
Tel: 972-3-537 5795, Fax: 972-3-537 5796
E-mail: aladdin@aladdin.co.il

United Kingdom — **Aladdin Knowledge Systems UK Ltd.**
Tel: 01753-622266, Fax: 01753-622262

France — **Aladdin France SA**
Tel: 1 40 85 98 85, Fax: 1 41 21 90 56

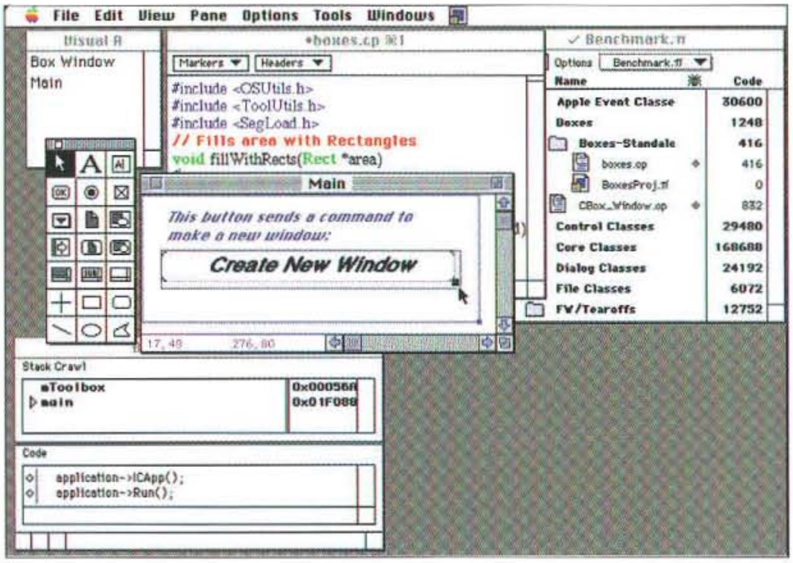
© Aladdin Knowledge Systems Ltd. 1985-1995 (6.95) HASP® is a registered trademark of Aladdin Knowledge Systems Ltd. All other product names are trademarks of their respective manufacturers. Mac & the Mac OS logo are trademarks of Apple Computer, Inc., used under license.

member of

■ **Aladdin Benelux** 08894 19777 ■ **Aladdin Japan** 0426 60 7191 ■ **Aladdin Russia** 095 9230588 ■ **Australia** Conlab 3 8985685 ■ **Czech** Atlas 2 766085
■ **Chile** Micrologica 2 222 1388 ■ **Denmark** Berendsen 39 577300 ■ **Egypt** Zeineldein 2 3604632 ■ **Finland** ID-Systems 0 870 3520 ■ **Germany** CSS 201 278804
■ **Greece** Unibrain 1 6856320 ■ **India** Solutions 11 2218254 ■ **Italy** Partner Data 2 26147380 ■ **Korea** Dae-A 2 848 4481 ■ **Mexico** SiSoft 5 5439770
■ **New Zealand** Training 4 5666014 ■ **Poland** Systherm 61 480273 ■ **Portugal** Futurmatica 1 4116269 ■ **Romania** Interactiv 64 153112
■ **South Africa** D Le Roux 11 886 4704 ■ **Spain** PC Hardware 3 4493193 ■ **Switzerland** Opag 61 7169277 ■ **Taiwan** Teco 2 555 9676 ■ **Turkey** Mikrobeta 312 467 7504

See Us At Comdex/Fall ! Las Vegas, NV, November 13-17, Booth S-7422

THE INDUSTRY STANDARD JUST MOVED TO A HIGHER POWER. SYMANTEC C++ 8.0 FOR POWER MAC.



```
#include <OSUtils.h>
#include <ToolUtils.h>
#include <SegLoad.h>
// Fills area with Rectangles
void fillWithRects(Rect *area)
```

Name	Code
Apple Event Classe	30600
Boxes	1248
Boxes-Standalc	416
boxes.cp	416
BoxesProj.π	0
CBox_Window.cp	832
Control Classes	29480
Core Classes	168688
Dialog Classes	24192
File Classes	6072
FW/Tearoffs	12752

```
 •Toolbox    0x00056A
▷ main       0x01F088
```

```
◇ application->ICApp();
◇ application->Run();
```

Boost your productivity with our Visual Architect, Advanced Editor, Debugger and Project Manager.

It's the highly-regarded industry standard, used by more developers than any other Macintosh development system. And now it's been totally re-engineered for Power Mac.

Introducing Symantec C++™ 8.0 for native Power Mac.

Not since the original THINK C™ has anything so dramatically boosted your productivity. There's a Visual Architect™ to generate GUI code instantly. An advanced Project Manager to handle the largest and most complex jobs. Plus native Power Mac tools for radically improved performance.

DRAW ON THE INDUSTRY STANDARD.

With Symantec C++ 8.0, you simply draw the user interface – including windows, dialogs, controls, icons and menus. Then the built-in Visual Architect generates the code with the click of a mouse. Now you can spend more time on what really sets your application apart – its functionality.

A HIGHER STANDARD FOR SPEED.

The new high-performance compiler is dramatically faster than the previous version, so you can become more productive than ever.

And for even more power, we've added an Advanced Project Manager. It gives you drag and drop so you can easily add files, Named Option Sets for changing complex sets of options fast, plus support for even the largest applications.

There's also a new editor and a browser for modifying and navigating source files, a new debugger, an incremental linker, THINK Class Library™ 2.0 and much more.

NEW! FAST C AND C++ COMPILERS
for both Mac and Power Mac

NEW! VISUAL ARCHITECT
for the Power Mac

NEW! THINK CLASS LIBRARY
2.0 *for Power Mac*

NEW! POPUP MENUS
take you right to declarations & headers

NEW! DEBUGGER
for nested projects and shared libraries

NEW! PROJECT MANAGER
for multiple targets and hierarchical support

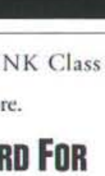
NEW! SPLIT PANE EDITOR
for syntax highlighting and auto formatting.

A DOUBLE STANDARD FOR 68K AND POWER MAC.

Symantec C++ 8.0 is the first native Mac/Power Mac development system to support C++ templates, nested classes and multiple inheritance, as well as ANSI C. Plus 8.0 includes version 7.0 for 68K support.

This double standard gives you everything you need for both 68K and Power Mac development.

To order at a special upgrade price of $149.95, call 1-800-628-4777 Ext. 9H21. Be sure to ask about the Symantec Developer's Advantage Program for premium support and regular updates. Or visit your local software store.



SYMANTEC.®

Price good in U. S. only. For more information in Canada, call 1-800-667-8661, ext. 5513. In Australia, call 2-879-6577. In Europe, call 31-71-353111. Symantec C++, Visual Architect, THINK Class Library are trademarks of Symantec Corporation. All other trademarks are the property of their respective holders. © 1995 Symantec Corporation. All rights reserved.

By Dave Mark, MacTech Magazine Regular Contributing Author

TCL and Visual Architect, Part 2

In last month's column, we took a look at Symantec's latest development environment, including the Symantec Project Manager and Visual Architect. Following the example presented in Chapter 14 of the *Symantec C++ User's Guide*, we used Visual Architect to generate source code to implement the single, default view named Main. We then returned to the Symantec Project Manager and compiled the code into a standalone application (See Figure 1).

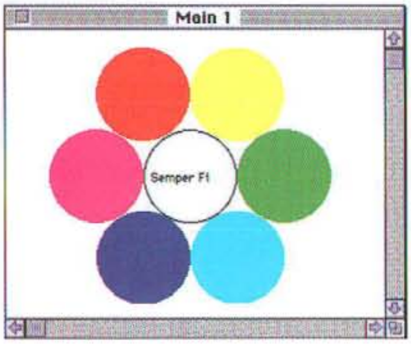


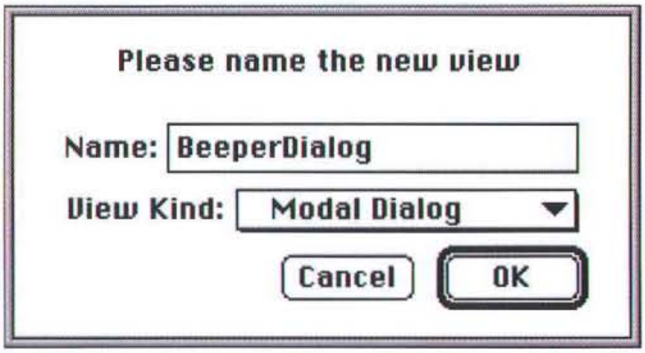
Figure 1. The window that appeared when we ran last month's program.

This month, we're going to add a second view to our program. This second view will be a dialog that allows you to type a number into an editable text field. When you press the **Beep** button, the dialog will beep the specified number of times.

Adding a Second View

Go into the Button *ƒ* folder you created last month and double-click on the file Button.π to launch the Symantec Project Manager. When the project window appears, double click on the file name Visual Architect.rsrc to bring up Visual Architect. This is where we'll add the second view.

- Select **New View** from the **View** menu.
- When asked to name the new view, type **BeeperDialog** and make sure **Modal Dialog** is selected from the **View Kind:** popup menu (Figure 2), then press the **OK** button.



Please name the new view

Name: BeeperDialog

View Kind: Modal Dialog

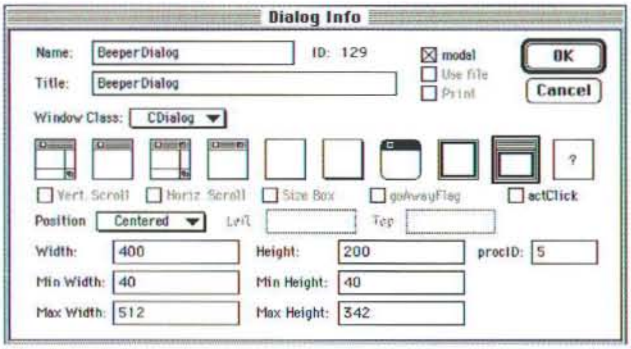
Figure 2. Naming the new view.

The name BeeperDialog will be used as the basis for a new class, called `CBeeperDialog`. The next time you select **Generate** from Visual Architect's Symantec Project Manager menu, Visual Architect will generate the files CBeeperDialog.cp, CBeeperDialog.h, x_CBeeperDialog.cp, and x_CBeeperDialog.h. First, we'll add a few items to this dialog and then add a button to our Main view that brings up the dialog.

Once you press the **OK** button, a view window will appear with the name **BeeperDialog**.

- Select **View Info** from the **View** menu.

A view info window titled **Dialog Info** will appear (Figure 3). The view info window, which looks a lot like a ResEdit or Resorcerer WIND templat, lets you customize the look of your new dialog window.



Name: BeeperDialog ID: 129
Title: BeeperDialog
☒ modal
□ Use file
□ Print
Window Class: CDialog
□ Vert. Scroll □ Horiz. Scroll □ Size Box □ goAwayFlag □ actClick
Position: Centered Left: Top:
Width: 400 Height: 200 procID: 5
Min Width: 40 Min Height: 40
Max Width: 512 Max Height: 342

Figure 3. The view info window for the BeeperDialog view.

- Click on the movable modal dialog icon (the second from the right) or type "5" in the procID: field.
- Click on the **OK** button to dismiss the dialog.
  Your next step is to add some items to the dialog.
- Click on the **Tools** menu and drag down until the tool palette outline appears.

When you release the mouse button, the Tools palette window will appear (Figure 4). Take a few minutes to play with the items in this palette. In the top row, you have the selection tool, the text tool, and the editable text tool. The second row features the pushbutton tool, the radio button tool, and the check box tool. The third row features the popup menu, icon, and PICT tools. We'll get into the rest of the tools in a future column. For now, try creating items based on the tools in the first three rows. Click on a tool, then click and drag in the BeeperDialog view window. Once you get a feel for these tools, select the selection tool (the arrow), then click on each of the items you added and press the delete key. Keep on deleting until the BeeperDialog window is empty again.

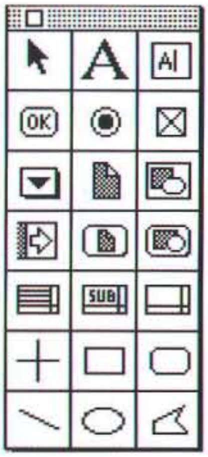


Figure 4. The Tools palette.

Now let's add the items that make up the Beeper dialog. We'll add four items: an **OK** button, a **Cancel** button, a static text label, and an editable text field for entering the number of beeps. We'll start with the **OK** and **Cancel** buttons.

- Click on the button tool in the palette window.
- Click on the BeeperDialog window and drag out an **OK** button.

As you drag, keep your eye on the lower left corner of the window. Notice that the two coordinates listed reflect the upper left and lower right corners of the button. I made my button 80 pixels wide and 20 pixels tall with an upper left corner of (200,42).

- Click on the BeeperDialog window and drag out a **Cancel** button.

My **Cancel** button was also 80 pixels wide and 20 pixels tall. This time, the upper left corner was (107,42). Notice that VA automatically filled in the text **OK** for the first button and **Cancel** for the second button. Nice feature! Let's change the **OK** to say **Beep**.

- Double-click on the **OK** button (or click on it and select **Pane Info...** from the **Pane** menu) to bring up the Pane Info window.
- Click on the triangle to the left of the class name **CControl**, revealing the CControl data members.
- Change contrlTitle from **OK** to **Beep** and close the Pane Info window.

Notice that the button's name changed from **OK** to **Beep**. Now let's add the static text label.

- Click on the static text tool (the big letter A) in the palette window.
- Click in the BeeperDialog window and type the text **Number of times to beep:**
- Click outside the static text to stop entering text, then click on the static text to select it. Drag it to a satisfactory position in the BeeperDialog window.

I dragged my static text so its upper-left corner was at (23,13). You might want to use the arrow keys to move the static text around the window, 4 pixels at a time. Hold down the command (⌘) key to get the arrow keys to move 1 pixel at a time. Experiment with the Shift and option keys which let you resize an item by 1 or 4 pixels in any of 4 directions.

Finally, let's add an editable text field.

- Click on the editable text tool (to the right of the static text tool) in the palette window.
- Click on the BeeperDialog window and drag out an editable text area.

Take a Power Trip to a New Realm of Developing

Powerful Tools to Take You Where You Want to Go

You've heard about the speed of PowerPC™ and you're thinking about the cool applications and content this amazing new chip will allow you to create. Read on to learn more about the tools that will take your development where you want to go, fast.

Prototyping HyperCard® 2.3 provides a robust prototyping and development environment for professional developers. It is also an easy-to-use tool for building a wide range of custom software applications for Macintosh®, from training simulations to information kiosks. And now it's accelerated for Power Macintosh®.

PowerPC and 68K Compilers Metrowerks CodeWarrior Gold is the industry's first native development environment for both the PowerPC and 68K-based Macintosh environments. You can easily target the entire Macintosh installed base with code created using one development environment running at Power Macintosh speeds.

Cross-Platform Multimedia Create innovative, interactive multimedia titles which will play on both Mac OS and Windows systems. Give users the ability to cruise up, down and around objects. They can even zoom in and pick them up. No wonder QuickTime™VR Authoring Tools Suite v.1.0 has won so many awards, including MacUser Magazine's "Eddy" for Breakthrough Technology of the Year.

There's a wide variety of development tools available through the Apple Developer Tools Catalog. Check it out. Developing on the Macintosh platform has never been easier or faster. And now with cross-platform flexibility, it's smarter than ever.

QuickTime VR Authoring Tools Suite v.1.0

Develop cross-platform multimedia products with QuickTime VR content

$495

This suite includes complete documentation for planning, designing, photographing and capturing content. QuickTime VR uses breakthrough compression/decompression techniques which produce small file sizes. And there's no special hardware required.

How to Order

To order call:

1 800 282-2732

And ask for a free copy of the Apple Developer Tools Catalog.
Visa, MasterCard and AMEX accepted.

Metrowerks CodeWarrior Gold

A complete development solution for C/C++ and Pascal programming.

$399

Runs native and cross-compiles on 68K or PowerPC platforms, ensuring that as you go forward on Power Macintosh you also conserve your investment in your installed code base on 68K Macintosh and your code base remains backward-compatible. Lets you compile, link, and debug 68K software at full native speed on a PowerMac™, and build PowerPC versions of your software on a 68K Mac.

HyperCard 2.3

A robust prototyping and development environment accelerated for Power Macintosh

$99

Comes with powerful scripting tools including AppleScript® Runtime built-in, a modeless script editor, hypertext support, debugging tools, new 24-bit color painting tools, Automated Button Tasks, and many other features. Stacks can be saved as stand alone, double-clickable applications you may distribute without software royalty fees.

© 1995 Apple Computer, Inc. Apple, the Apple logo, Macintosh, Power Macintosh, HyperCard, AppleScript and QuickTime are registered trademarks of Apple Computer, Inc. PowerMac is a trademark of Apple Computer, Inc. PowerPC is a trademark of International Business Machines Corporation under license therefrom. Offer expires 12/31/95. Prices above do not include tax and shipping. Prices and products are subject to change without notice. To place orders from Canada call 1-800-637-0029; outside Canada call 1-716-871-6555 or fax inquiries to 1-716-871-6511.

MT1095

Mine had and upper left corner of (205,13) and a lower right of (280,29). Once your items are in place, resize the dialog view so it comfortably encloses the dialog items. To do that, click on the handle in the lower-right corner of the grey rectangle embedded in the BeeperDialog window. Drag the handle to its new position. I dragged mine to (293,75). The finished Beeper dialog is shown in Figure 5.

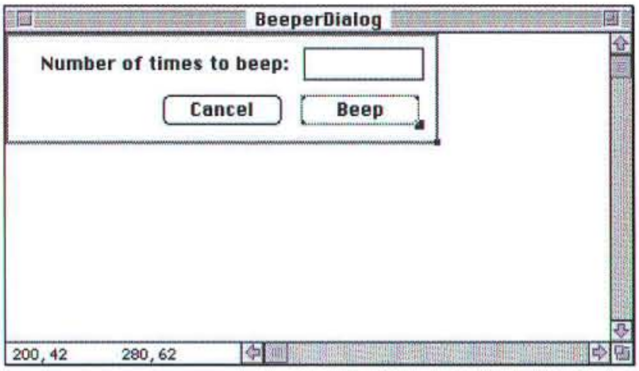


Figure 5. The finished Beeper dialog with the **Beep** *button selected.*

If you like, you can try the dialog out by selecting **Try Out** from the **View** menu. Click on one of the buttons or select **Close** from the **File** menu to close the Try Out window once you are done admiring your handiwork.

Our next task is to tell VA that our edit-text field is designed to hold only integers (as opposed to any old text).

- Click on the editable text field.
- Select **Class** from the **Pane** menu, then select **CIntegerText** from the submenu that appears.

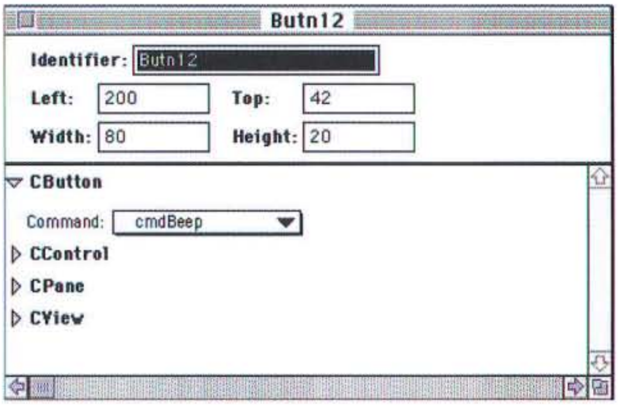
Next, we need to define a new command that will get sent when the **Beep** button is clicked. Right now the **Beep** button's command is set to cmdOK. To see that, double-click on the **Beep** button to bring up the Pane Info window, then click on the **CButton** triangle. Take a look at the Command popup menu. If you click on it, you'll find a whole bunch of commands, all of them built into the TCL. Since there is no Beep command, we'll need to define it ourselves, then set the **Beep** button to send it when clicked.

- Close the **Beep** Pane Info window, if you opened it to look at the Command popup.
- With the BeeperDialog window frontmost, select **Commands...** from the **Edit** menu.
- When the Commands window appears, select **New Command** from the **Edit** menu.
- Type **cmdBeep** in the uppermost editable text field.
- Select **CBeeperDialog** from the **In Class:** popup menu. This tells VA which class this command belongs to.
- Select **Call** from the **Do:** popup menu.

This tells VA what you want done in response to this command. You could do nothing, you could ask VA to call a function, or you could open another view. In our case, we've asked VA to call a CBeeperDialog member function. We'll get to that function in a bit.

Notice that VA automatically generated the command number 512. This is not particularly important. You'll use the constant cmdBeep instead of 512, so there's no reason to memorize your command numbers.

- Click on the **OK** button to dismiss the Commands dialog. As you've probably guessed, your next step is to associate the cmdBeep command with the **Beep** button.
- Double-click on the **Beep** button to bring up the Pane Info window.
- Click on the triangle to the left of the class **CButton**, revealing the **Command** popup menu.
- Select **cmdBeep** from the **Command** popup (Figure 6).
- Close the Pane Info window.



Identifier: Butn12
Left: 200 Top: 42
Width: 80 Height: 20
Command: cmdBeep

Figure 6. The Pane Info window associated with the **Beep** *button. Notice that the Command popup has been set to cmdBeep.*

You may have noticed that the **Beep** button no longer has the bold, rounded rectangle around it, specifying it as the default button, the button pressed when the user hits return. This changed because VA specifies the cmdOK command as the default. Let's fix that.

- Select **Set Default Command** from the **View** menu.
- Select **cmdBeep** from the **Command** popup menu, then press **OK**.

Notice that the bold, rounded rectangle is back around the **Beep** button (Figure 7).

Announcing

MICROGUARD PLUS™

Why so many developers are switching to MicroGuard copy protection

MicroGuard is committed to uncompromising technological superiority

"Technology at its peak" is our commitment to you. That is why we have brought you **MicroGuard Plus™**. And, MicroGuard Plus is **100% backwards** compatible with MicroGuard. This means MicroGuard and MicroGuard Plus can be used interchangeably. Just as we promised!

MicroGuard offers you the most sophisticated network protection

Our network protection, **MicroGuard Net™**, is so superior, we had to hire an Apple network engineer to execute our specifications.

MicroGuard is the only key developed by Mac developers, and is the first and only 100% ADB savvy key

We have been developing Mac applications as a seed development house since 1984. We are not a PC protection company that has come to you with a Mac product. MicroGuard is fully ADB savvy and offers extended addressing. Unlike other protection devices, MicroGuard never clashes with other keys. Only MicroGuard offers this level of sophistication.

MicroGuard has now surpassed its own technological lead, actually improving on the best

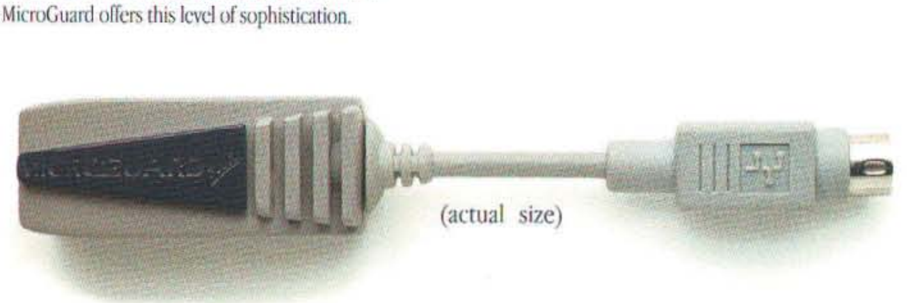
MicroGuard Plus is everything MicroGuard is, plus 40-bit encryption, two additional passwords, 64-Bit Array, 32-byte public area, 45% size reduction, enhanced counter and more. In addition, MicroGuard Plus offers two new utilities: **QuickGuard™** and EasyGuard™ allow you to protect your applications without touching your source code. The only feature that is not plus is the price. :-)

MicroGuard is the best selling Macintosh key in the world

MicroGuard sells more Macintosh copy-protection keys than anyone else in the world!

MicroGuard delivers developer support within 24 hours

We will answer any inquiry you have within 24-hours. We also have a fully loaded AppleLink bulletin board which contains all our libraries, tech notes, Q&A and nearly everything you'll ever need!



(actual size)

For more information and to order a Developer's Kit or to receive a free CD ROM about MicroGuard, please contact us at:

MicroGuard USA: Tel: (303) 320-1628 • Fax: (303) 320-1599 • AppleLink: M.GUARD
International: Tel: (972) 3 558-2345 • Fax: (972) 3 558-2344 • AppleLink: MICROGUARD

Mac™OS

Pirate Busters

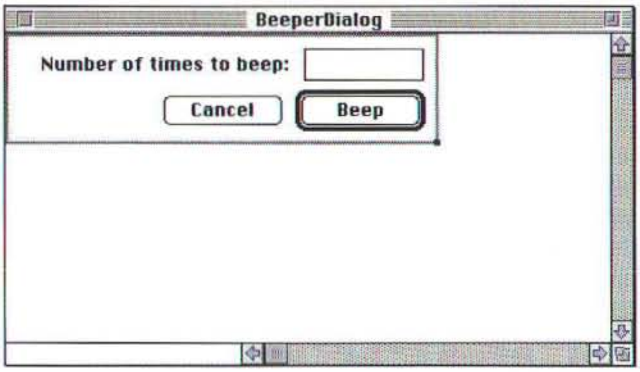


Figure 7. The **BeeperDialog** *window, once cmdBeep is set as the default command.*

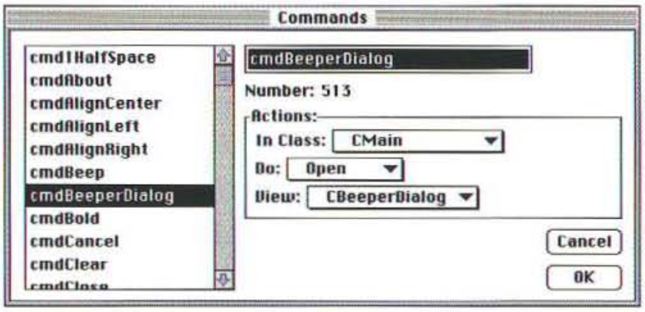
We are almost done! Now we have to add a button to the Main view that brings up the Beeper dialog.

- Close the BeeperDialog view window.
- Double-click on Main in the view list window to bring up the Main view window.
- Click on the grow box in the lower left corner of the Main view window and drag down about 50 pixels.
- Click on the handle in the lower right corner of the grey bounding rectangle inside the window and drag it down about 50 pixels.

You've just resized the Main view to be about 50 pixels taller, making room for a new pushbutton.

- Click on the pushbutton tool (under the arrow) in the palette window.
- Click in the Main view window and drag out a new button below the PICT.
- Before you click off the button, type the word **Beeper...** The button will now bear the name **Beeper...**.
- Double-click on the **Beeper...** button, bringing up the button's Pane Info window.
- Click on the triangle next to the class name CButton to reveal the CButton data members.
- Choose **Other...** from the top of the **Command** popup menu.
- When the **Commands** dialog appears, choose **New Command** from the **Edit** menu.
- Enter the command name **cmdBeeperDialog**.
- Choose **CMain** from the **In Class:** popup menu to let VA know you want the new command associated with the CMain class.
- Choose **Open** from the **Do:** popup menu.
- Choose **CBeeperDialog** from the **View:** popup menu.

We've just told VA that we want this new command (cmdBeeperDialog) to open the CBeeperDialog view (See Figure 8).


Commands

cmd1HalfSpace
cmdAbout
cmdAlignCenter
cmdAlignLeft
cmdAlignRight
cmdBeep
cmdBeeperDialog
cmdBold
cmdCancel
cmdClear
cmdClose

cmdBeeperDialog
Number: 513
Actions:
In Class: CMain
Do: Open
View: CBeeperDialog

Figure 8. The Commands window for the new cmdBeeperDialog command.

- Click on the **OK** button.

Now we've told VA to associate this command with the **Beeper...** button we just created.

- Close the button's Pane Info window to return to the Main view window (Figure 9).

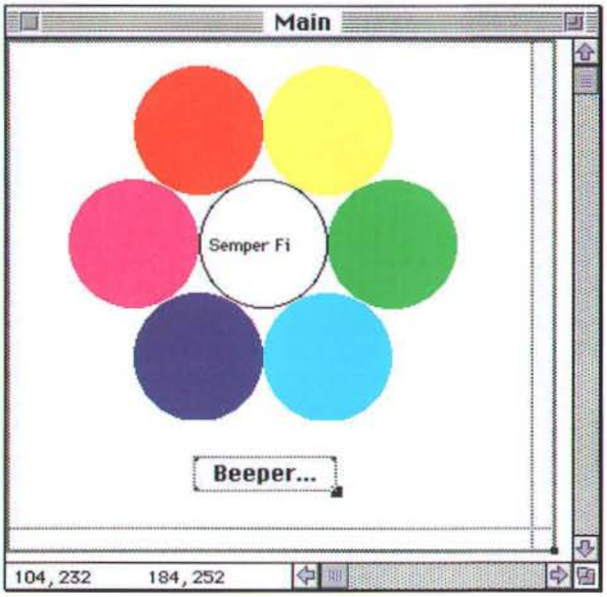


Figure 9. My Main view with my new **Beeper...** *button.*

If you like, select **Try Out** from the **View** menu and take your new version of Main for a spin. Note that we haven't put the code in place that gets called when the **Beeper...** button is pressed, but the scrollbars should work and the button should highlight when clicked.

Our next step is to ask VA to regenerate our source code, reflecting the new panes and commands we've defined.

- Select **Generate...** from the Symantec Project Manager menu (diamond shaped menu to the right of the **Windows** menu).

- When prompted to save your changes, click **OK**.
  VA will update your existing code base. If you selected **Generate All...**, VA would have replaced your existing code base. Be sure you understand this difference!

Adding the New Code

Our last step is to add a few lines of code to bring all our new interface elements together.

- Return to the Symantec Project Manager.
- In the project window, click on the triangle to the left of the folder labeled Source. Skip this step if the Source folder was already opened.
- Double click on the file name "CBeeperDialog.cp".
- Add this code to the end of CBeeperDialog.cp:

```
void CBeeperDialog::DoCmdBeep()
{
  long n;
  n = fBeeperDialog_Edit15->GetIntValue();
  for ( long i=0; i<n; i++ )
    SysBeep( 20 );
}
```

This code references the data member fBeeperDialog_Edit15. When VA generates your code, it makes up a constant for each of your dialog items. You can find these definitions in the file BeeperDialogItems.h. Here's the constants that were generated for my version of Beeper:

```
enum
{
  BeeperDialog_Begin_,
  kBeeperDialog_Butn12 = 1,
  kBeeperDialog_Butn12ID = 2049L,
  kBeeperDialog_Butn13 = 2,
  kBeeperDialog_Butn13ID = 2050L,
  kBeeperDialog_Stat14 = 3,
  kBeeperDialog_Stat14ID = 2051L,
  kBeeperDialog_Stat17 = 4,
  kBeeperDialog_Stat17ID = 2053L,
  kBeeperDialog_Edit15 = 5,
  kBeeperDialog_Edit15ID = 2052L,
  BeeperDialog_End_
};
```

The member function `GetIntValue()` retrieves the value from the field that calls it (in this case, the editable text field), translates the value into an integer, and returns it. The loop uses this value to beep the specified number of times.

- Go to the top of the file and remove the comment characters (//) at the beginning of the line:

```
//#include "AppCommands.h"
```

You'll need to do this last step whenever your view includes at least 1 command.

- Close CBeeperDialog.cp.
- Open the file CBeeperDialog.h and add this member function declaration to the public section of the CBeeperDialog class:

```
void DoCmdBeep();
```

That's it! When you run the program, the Main view should appear, featuring a brand new button (Figure 10).

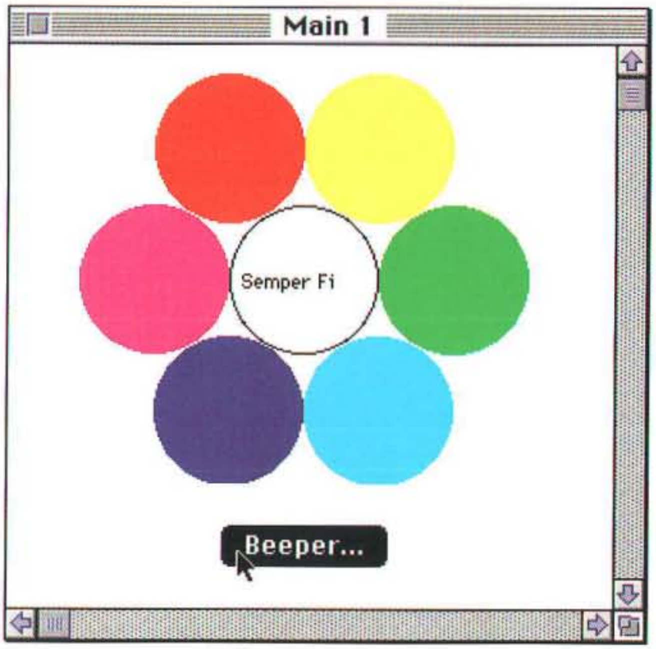
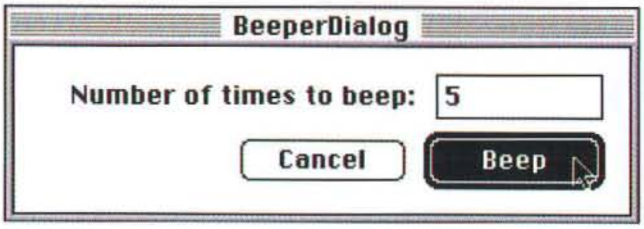


Figure 10. The Main view, with the **Beeper...** *button pressed.*

Press the **Beeper...** button. The Beeper Dialog will appear. Type a number in the text field and press the **Beep** button. Happy?



Number of times to beep: 5

Figure 11. Pressing the **Beep** *button.*

Till Next Month...

Next month, we'll go back to PowerPlant. I just got my copy of CW7, and I'm anxious to dig into it. Hope you got your upgrades, too. See you next month...

Visit MacTech Magazine's Web site!
http://www.mactech.com

Developer Central™

Sponsored by

and

For Macintosh Programmers & Developers

MacTech™ MAGAZINE

Macworld Expo/San Francisco • January 9 – 12, 1996
Moscone Convention Center

Come to the Center of the Universe!

(The Macintosh development universe.)

If you are developing custom applications, custom solutions, multimedia, or engineering solutions for the Macintosh, then you need to visit Developer Central. Here – all in one place – you will see the latest tools for Macintosh and Newton development. In addition, Apple Computer will be available with information on its training and programs to support developers.

If you want to jump-start your knowledge of Macintosh development or come up to speed on a new technology or tool, then visit the Developer Central Theater, where representatives of Apple's Developer University and others will be presenting the inside scoop on Macintosh development.

At Developer Central, you can…

- Talk to representatives and demo products from a number of Third Party Vendors
- Attend sessions in the Developer Central Theater
- Get the best deals on tools from APDA and MacTech

So, no matter if you are corporate developer putting together a departmental system,
use the Macintosh in science,
or are aiming to have the next great Macintosh application or multimedia title,
you will find what you need at Developer Central.

Developer Central
The Source for Macintosh Development

Sponsored by Apple Computer, Inc. and MacTech Magazine

Apple and the Apple logo are trademarks of Apple Computer, Inc. MacTech and MacTech Magazine are trademarks of Xplain Corporation.
Developer Central and Virtual Developer Central are trademarks of Xplain Corporation and Apple Computer, Inc.

PROGRAMMER'S CHALLENGE

By Bob Boonstra, Westford, Massachusetts

Enclosing Bounds

The Challenge this month is based on a suggestion by Mike Scanlin, who remains a fan of the column. (We're still waiting for Mike's first Challenge entry, however.) The problem is to write a routine that will return a rectangle enclosing all non-white pixels in a selected area of an image. This code might be useful in a drawing or painting program, where the user would be allowed to select a subset of the image by clicking and dragging, and the software would select all of the elements of the image contained within that selection. The prototype of the code you will write is:

```
void EnclosingBounds(
  PixMapHandle pm,          /* handle to PixMap containing image */
  Rect selection,           /* subset of image to enclose */
  Rect *enclosingRect       /* enclosing rect return value */
);
```

Your code should examine all of the pixels within the `selection` rectangle of the PixMap and return the smallest rectangle containing all of the non-white pixels. Pixels outside the `selection` rectangle should be ignored. The `bounds` rectangle of the PixMap will be no larger than 2048 pixels in each dimension, the `baseAddr` pointer will be longword aligned, and `rowBytes` will be a multiple of 4. You should deal with `pixelSize` values of 1, 8, or 32, with values of 8 and 32 being weighted most heavily in measuring performance. For PixMaps with indexed pixels (`cmpCount==1`), the color table will contain white as the first table entry (as all good color tables are supposed to). For PixMaps with direct pixels, the unused (alpha) bits of each pixel will be zero.

You may use either the Metrowerks or the Symantec compilers for this native PowerPC Challenge. If you have any questions, or would like some test data for your code, please send me e-mail at one of the Programmer's Challenge addresses, or directly to boonstra@ultranet.com.

Two Months Ago Winner

Congratulations to **Eric Lengyel** (Blacksburg, VA) for submitting the fastest and smallest entry to the Reversible Scrambling Algorithm Challenge. Despite an unfortunate delay in publication of the magazine that left participants with less time than usual to complete the Challenge, three of the four entries I received by the extended deadline worked correctly, at least in part.

You might recall that the Challenge was to write code that would raise a large integer message to a power and compute the remainder modulo another large integer. The name of the Challenge comes from the fact that this technique is reversible, given properly chosen integers. Eric is a graduate student in Mathematics at Virginia Tech, and he took advantage of a highly optimized multiple precision integer arithmetic library that he had written as part of a number theory project involving the factorization of very large numbers.

Each of the working entries converted the BigNum representation provided in the problem into one that right-justified numbers into a fixed-length data structure. While this imposes a restriction on the maximum size integer that the code can handle, this assumption was permitted by the problem statement. In Eric's code, the restriction is controlled by a single `#define` statement.

Eric uses a binary exponentiation algorithm to raise the message to the specified power, and takes advantage of facts from number theory that allow the remainder to be computed at each step of the exponentiation. The time to perform the exponentiation is therefore proportional to the logarithm of the

THE RULES

Here's how it works: Each month we present a new programming challenge here. First, write some code that solves the challenge. Second, optimize your code (a lot). Then, submit your solution to MacTech magazine. We choose a winner based on code correctness, speed, size, and elegance (in that order of importance) as well as the submission date. In the event of multiple equally desirable solutions, we'll choose one winner (with honorable mention, but no prize, given to the runner up). The prize for each month's best solution is a $100 credit in the MacTech Mail Order Store and a limited-edition, "The Winner! MacTech Programmers Challenge" T-shirt (not available in stores anywhere).

Unless stated otherwise in the problem statement, the following rules apply: All solutions must be in ANSI compatible C. Use only pure C code. We disqualify entries with any assembly in them (except for challenges specifically stated otherwise). You may call any Macintosh Toolbox routine (e.g., it doesn't matter if you use NewPtr instead of malloc). We test entries with compiler options set to disable FPU use (for 680x0 code) and to enable all available speed optimizations. The compiler to be used and the target instruction set (680x0 or PowerPC) will be stated in the problem. **Limit your code to 60 characters per line;** this helps with e-mail gateways and page layout.

We publish the solution and winners for this month's Programmer's Challenge two months later. All submissions must be ***received by*** the 10th day of the month printed on the front cover of this issue.

You can get a head start on the challenge by reading the online version. We post it to the online services at the same time that we post source code. We make every effort to have it online no later than when the magazines are mailed, but we're unable to guarantee that it will be online by any given date.

Mark solutions "Attn: Programmer's Challenge Solution" and send it by e-mail to one of the Programmer's Challenge addresses in the "How to Communicate With Us" section on page 2 of this issue. Include the solution, all related files, and your contact info.

MacTech Magazine reserves the right to publish any solution entered in the Programmer's Challenge. Authors grant MacTech Magazine the exclusive right to publish entries without limitation upon submission of each entry. Authors retain copyrights for the code.

Get Net Wise... NOW!

JointSolutions Marketing is now creating exciting World Wide Web pages for various clients, including Apple Computer, Inc.

We can also assist you with all aspects of database publishing and interface design.

Call for the Web addresses of our latest projects.

Don't miss this opportunity to reach your target audience. Call today!

- ***Web Page Design***
- ***html Conversion***
- ***Button and Screen Design***
- ***Interface Design***
- ***Database Publishing***
- ***Project Management***

JointSolutions Marketing
408/338-6471
408/338-6475 (fax)
jsolver@aol.com (e-mail)

JOINT SOLUTIONS MARKETING

exponent. Eric's multiplication and division routines use the 68020's capability to compute the 64-bit product of two longwords and to divide a 64-bit dividend by a longword. The multiplication, division, exponentiation, and compare routines in Eric's code are general purpose and could be used in any 68K application that needs large integers.

Honorable mention goes to Ernst Munter, who submitted an entry in pure C that was actually the fastest code for the short modulus test cases. Unfortunately, his entry did not produce correct results for the longer moduli.

Here are the times and code sizes for the entries that worked correctly (or partially correctly). Execution time is presented for two specific test cases, with modulus lengths of 22 and 88 bytes, respectively, as well as the total time for all of the test cases I ran. Cases that produced incorrect results are indicated with an asterisk. Numbers in parens after a person's name indicate that person's cumulative point total for all previous Challenges, not including this one.

Name	time1	time2	Total time	code	data
	(22 bytes)	(88 bytes)			
Eric Lengyel	47	463	2083	1190	0
Xan Gregg (51)	35	967	3175	1558	0
Ernst Munter (C entry) (90)	17	*	*	4266	11788

Top 20 Contestants of All Time

Here are the Top 20 Contestants for the Programmer's Challenges to date. The numbers below include points awarded for this month's entrants. (Note: ties are listed alphabetically by last name – there are more than 20 people listed this month because of ties.)

Rank	Name	Points
1.	[Name deleted]	176
2.	Munter, Ernst	90
3.	Karsh, Bill	78
4.	Stenger, Allen	65
5.	Gregg, Xan	61
6.	Larsson, Gustav	60
7.	Riha, Stepan	51
8.	Goebel, James	49
9.	Nepsund, Ronald	47
10.	Cutts, Kevin	46
11.	Mallett, Jeff	44
12.	Kasparian, Raffi	42
13.	Vineyard, Jeremy	42
14.	Darrah, Dave	31
15.	Landry, Larry	29
16.	Elwertowski, Tom	24

17.	Lee, Johnny	22
18.	Noll, Robert	22
19.	Anderson, Troy	20
20.	Beith, Gary	20
21.	Burgoyne, Nick	20
22.	Galway, Will	20
23.	Israelson, Steve	20
24.	Landweber, Greg	20
25.	Lengyel, Eric	20
26.	Pinkerton, Tom	20

There are three ways to earn points: (1) scoring in the top 5 of any Challenge, (2) being the first person to find a bug in a published winning solution or, (3) being the first person to suggest a Challenge that I use. The points you can win are:

1st place	20 points
2nd place	10 points
3rd place	7 points
4th place	4 points
5th place	2 points
finding bug	2 points
suggesting Challenge	2 points

Here is Eric's winning solution:

PowerAndRemainder.c

```
Copyright © 1995 Eric Lengyel
/*
I call my fixed length numbers "BigFixed" and translate from BigNum's to BigFixed's in
the PowerAndRemainder routine.  These are the assembly language routines which are
the guts of my program:

(1) PowerMod - raises a number to a power and reduces it by a modulus.  It uses a fast
binary exponentiation algorithm, reducing by the modulus at each step.
(2) Multiply - multiplies 2 BigNum's together.
(3) MultQ - mutliplies a BigNum by a long int.
(4) Divide - divides one BigNum by another and supplies the quotient and remainder.
(5) Compare - determines the ordering of 2 BigNum's.

Some of the loops have been expanded to make more efficient use of the instruction
cache.
*/

#define NumSize 72

typedef struct BigNum
{
   short          numDig;
   unsigned char  *dig;
} BigNum;

typedef struct BigFixed
{
   unsigned char  dig[NumSize*4];
} BigFixed;

/* We need 72 longs because the division routine needs the most significant longword
to be zero and the speed optimization requires that NumSize be a multiple of four. */

void PowerAndRemainder(BigNum *msg, BigNum *exp, BigNum *n,
   BigNum *res);
void PowerMod(BigFixed *msg, BigFixed *exp, BigFixed *n,
   BigFixed *res);
void Multiply(BigFixed *src1, BigFixed *src2, BigFixed *dst);
void MultQ(BigFixed *src1, long src2, BigFixed *dst);
void Divide(BigFixed *end, BigFixed *sor, BigFixed *dst);
short Compare(BigFixed *src1, BigFixed *src2);
```

PowerAndRemainder

```
void PowerAndRemainder(BigNum *msg, BigNum *exp, BigNum *n,
   BigNum *res)
{
   short       a, b, numDigits;
   BigFixed    msg0, exp0, n0, res0;

   for (a = 0; a < NumSize*4; a++)
   {
      b = NumSize*4 - msg->numDig;
      if (a < b) msg0.dig[a] = 0;
      else msg0.dig[a] = msg->dig[a - b];
      b - NumSize*4 - exp->numDig;
      if (a < b) exp0.dig[a] = 0;
      else exp0.dig[a] = exp->dig[a - b];
      b = NumSize*4 - n->numDig;
      if (a < b) n0.dig[a] = 0;
      else n0.dig[a] = n->dig[a - b];
   }
   PowerMod(&msg0, &exp0, &n0, &res0);
   a = 0;
   while (res0.dig[a] == 0) a++;
   numDigits = res->numDig = NumSize*4 - a;
   for (b = 0; b < numDigits; b++)
      res->dig[b] = res0.dig[a++];
}
```

PowerMod

```
void PowerMod(BigFixed *msg, BigFixed *exp, BigFixed *n,
   BigFixed *res)
{
   BigFixed   acc, scrap;

   asm
   {
   LEA       acc, A0              ; Start with one in
   MOVEQ     #NumSize/4-2, D0     ; accumulator
@Z CLR.L     (A0)+
   CLR.L     (A0)+
   CLR.L     (A0)+
   CLR.L     (A0)+
   DBF       D0, @Z
   CLR.L     (A0)+
   CLR.L     (A0)+
   CLR.L     (A0)+
   MOVEQ     #1, D0
   MOVE.L    D0, (A0)
   MOVEA.L   exp, A0              ; Put address of
                                  ; exponent in A0
   CLR.W     D2                   ; D2 holds position
                                  ; in exponent
@6 TST.L     00(A0, D2.W*4)       ; Find first longword
                                  ; of exponent
   BNE       @4                   ; which is not zero
   CMPI.W    #NumSize-1, D2
   BEQ       @5                   ; If entire exponent
                                  ; is zero, leave
   ADDQ.W    #1, D2
   BRA       @6
@4 CLR.L     D1                   ; D1 holds the bit #
@2 PEA       acc                  ; Square accumulator
   PEA       acc
   PEA       acc
   JSR       Multiply
   ADDA.W    #12, A7
   MOVE.L    n, -(A7)             ; Compare accumulator
   PEA       acc                  ; to modulus
   JSR       Compare
   ADDQ.W    #8, A7
   TST.B     D0
   BMI       @7                   ; If it's less, skip
                                  ; reduction
   PEA       scrap                ; Reduce accumulator
                                  ; modulo "n"
   MOVE.L    n, -(A7)             ; Quotient not needed,
   PEA       acc                  ; store in "scrap"
   JSR       Divide
   ADDA.W   #12, A7
@7 BFTST    00(A0, D2.W*4){D1:1}  ; Test a bit in current
                                  ; longword of exponent
   BEQ      @1                    ; If zero, skip multiply
   PEA      acc                   ; Multiply accumulator
   PEA      acc                   ; by base
   MOVE.L   msg, -(A7)
   JSR      Multiply
   ADDA.W   #12, A7
   MOVE.L   n, -(A7)              ; Compare accumulator
   PEA      acc                   ; to modulus
   JSR      Compare
   ADDQ.W   #8, A7
   TST.B    D0
   BMI      @1                    ; If it's less, skip
                                  ; reduction
   PEA      scrap                 ; Reduce modulo "n"
   MOVE.L   n, -(A7)
   PEA      acc
   JSR      Divide
   ADDA.W   #12, A7
@1 ADDQ.W   #1, D1                ; Increase bit number
   CMPI.B   #32, D1
   BLT      @2                    ; If < 32, loop
   ADDQ.W   #1, D2                ; Increase longword #
   CMPI.W   #NumSize, D2
   BLT      @4                    ; If < number size, loop
@5 LEA      acc, A0               ; Copy acc to "res"
   MOVEA.L  res, A1
   MOVEQ    #NumSize/4-1, D0
@3 MOVE.L   (A0)+, (A1)+
   MOVE.L   (A0)+, (A1)+
   MOVE.L   (A0)+, (A1)+
   MOVE.L   (A0)+, (A1)+
   DBF      D0, @3
   }
}
```

Multiply

```
/* Multiply src1 by src2 and put product in dst */

void Multiply(BigFixed *src1, BigFixed *src2, BigFixed *dst)
{
   short      topStop, botStop;
   BigFixed   acc, line;

   asm
   {
   MOVEM.L  D0-D7/A0-A4, -(A7)
   LEA      acc, A0               ; Clear accumulator
   MOVEQ    #NumSize/4-1, D0
@Z CLR.L    (A0)+
   CLR.L    (A0)+
   CLR.L    (A0)+
   CLR.L    (A0)+
   DBF      D0, @Z
   LEA      line, A2              ; "line" holds partial
                                  ; products
   MOVEA.L  src1, A3              ; Move bottom number's
   CLR.W    D1                    ; address into A3
@6 TST.L    00(A3, D1.W*4)        ; Find first non-zero
   BNE      @7                    ; longword of bottom #
   ADDQ.W   #1, D1
   CMPI.W   #NumSize, D1          ; If bottom number 0,
   BEQ      @8                    ; go to end
   BRA      @6                    ; Loop
@7 MOVE.W   D1, botStop
   MOVEQ    #NumSize - 1, D4      ; D4 holds position in
                                  ; bottom number
@4 MOVEA.L  src2, A4              ; Move top number's
   CLR.W    D1                    ; address into A4
@9 TST.L    00(A4, D1.W*4)        ; Find first non-zero
   BNE      @A                    ; longword of top #
   ADDQ.W   #1, D1
   CMPI.W   #NumSize, D1          ; If top number zero,
   BEQ      @8                    ; go to end
   BRA      @9                    ; Loop
@A SUBQ.W   #1, D1
   BPL      @X
   CLR.W    D1
@X MOVE.W   D1, topStop
@B CLR.L    00(A2, D1.W*4)        ; Clear leading longwords
   DBF      D1, @B                ; of partial product
   ADDA.W   #NumSize * 4, A4      ; Add size
   CLR.L    D2                    ; D2 is carry register
   MOVE.L   00(A3, D4.W*4), D7    ; Get longword from
                                  ; bottom number
   MOVEQ    #NumSize - 1, D3      ; D3 holds position
                                  ; in product
@1 MOVE.L   D7, D5                ; Copy longword to D5
   MULU.L   -(A4), D6:D5          ; Do 64-bit multiply
   ADD.L    D2, D5                ; Add carry to low
                                  ; longword of product
   CLR.L    D2                    ; Use D2 as dummy to
                                  ; extend carry
   ADDX.L   D2, D6                ; Add zero to high
                                  ; longword with carry
   MOVE.L   D6, D2                ; Anything in high
                                  ; longword gets carried
   MOVE.L   D5, 00(A2, D3.W*4)    ; Store low longword in
                                  ; partial product
   SUBQ.W   #1, D3                ; Loop through all
   CMP.W    topStop, D3           ; longwords in top number
   BGE      @1
   MOVEA.L  A2, A0                ; Now add partial product
                                  ; to accumulator
   MOVE.L   D4, D0                ; Calculate correct
                                  ; position in product
   LEA      acc, A1               ; Get accumulator's addr
   ADDQ.W   #1, D0
   ADDA.W   #NumSize * 4, A0
   LSL.W    #2, D0
   ADDA.W   D0, A1
   MOVE.W   D4, D1
   MOVE.L   -(A1), D0             ; Get longword of product
   SUBQ     #1, D1
   ADD.L    -(A0), D0             ; Add longword of
   MOVE.L   D0, (A1)              ; partial product
   TST.W    D1                    ; If no more longwords,
   BMI      @2                    ; then branch
@3 ADDX.L   -(A0), -(A1)          ; Add next longword
   DBF      D1, @3                ; Loop through
                                  ; entire product
@2 SUBQ.W   #1, D4                ; Loop and do next
   CMP.W    botStop, D4           ; longword of bottom #
   BGE      @4
@8 LEA      acc, A0               ; Copy product to "dst"
   MOVEA.L  dst, A1
   MOVEQ    #NumSize/4-1, D0
@5 MOVE.L   (A0)+, (A1)+
   MOVE.L   (A0)+, (A1)+
   MOVE.L   (A0)+, (A1)+
   MOVE.L   (A0)+, (A1)+
   DBF      D0, @5
   MOVEM.L  (A7)+, D0-D7/A0-A4
   }
}
```

MultQ

```
/* Multiply src1 by src2 and put product in dst */

void MultQ(BigFixed *src1, long src2, BigFixed *dst)
{
   BigFixed   pro;

   asm
   {
   MOVEM.L  D0-D7/A0/A1, -(A7)
   LEA      pro, A0               ; Clear product
   MOVEQ    #NumSize/4-1, D0
@Z CLR.L    (A0)+
   CLR.L    (A0)+
   CLR.L    (A0)+
   CLR.L    (A0)+
   DBF      D0, @Z
   LEA      pro, A1               ; Get address of
                                  ; product
   MOVEA.L  src1, A0              ; Move top number's
   CLR.W    D1                    ; address into A0
@3 TST.L    00(A0, D1.W*4)        ; Find first non-zero
   BNE      @4                    ; longword of top #
```

Source Surfer.

Introducing the incoming wave in Mac development tools: BBEdit 3.5. This radical text editor provides an amazing variety of ways to browse, search, and surf through your projects.

Let's Go Surfing.

BBEdit's integrated disk browser provides you with a quick and easy means of navigating any mounted volume. It's the ideal tool for exploring developer and E.T.O. CDs. BBEdit also has the unique ability to open multiple THINK or CodeWarrior project documents, either for browsing or searching.

Can't Find It? Just Ask.

At the heart of BBEdit's surfing ability is a powerful engine for searching text, either literally or with UNIX™-style "grep" patterns. This search engine is coupled to multi-file directory scanning, and the ability to search currently open windows, previously searched files, or project documents (even if your development environment isn't running). The new "Quick Search" capability gives interactive searching as you type. So even if you don't know what you're looking for, BBEdit can find it.

Project Your Feelings.

All of this great capability would be wasted if you couldn't use BBEdit to write code. That's why we've incorporated awareness of Projector, direct support for ToolServer, and seamless integration as an external editor for use with THINK C/Symantec C++ (version 6.0 and later), as well as support for control of Metrowerks CodeWarrior (version 1.2 and later).

We Beg To Diff.

BBEdit's unique ability to compare folders and projects makes it a natural for figuring out which changes you made at 3 AM the night before.

We caught the wave.

In addition to its renowned surfing abilities, BBEdit supports the latest system software technologies: Apple Guide, AppleScript, Macintosh Drag and Drop, PowerTalk, PowerPC acceleration, and Quickdraw GX printing. There are goodies to make your life easier: "soft" text wrapping, automatic linefeed translation, parenthesis balancing, auto-indent, and more. You can also extend BBEdit with plug-in code modules. (We provide information and examples to help.)

We *are* the wave.

Since its first commercial release in 1993, BBEdit has set the pace for capability, user interface, functionality, and customer support that others are still trying to emulate. BBEdit was the first editor to support ToolServer, THINK Reference, Toolbox Assistant, THINK C/Symantec C++, and the leading-edge technologies mentioned above. We're still setting the pace for others to follow, as the first Mac text-editing solution to deliver on CD-ROM with on-board documentation, the first text editor to support the Internet Config system, and the first development tool with Apple Guide.

"I Tried it Once, But Didn't Inhale."

We've painted a great picture, but don't take our word for it. Take a deep breath, hit our FTP site (see below) and see for yourself. You'll find lots of things to like.

BBEdit. It doesn't suck.

Bare Bones
Software, Inc.

For more information:

Internet: bbsw@netcom.com
CompuServe: 73051,3255
Applelink: BARE.BONES
eWorld: BareBones
ftp://ftp.netcom.com/pub/bb/bbsw/

Accelerated for
Power Macintosh

Mac™OS

We also have "It doesn't suck" t-shirts that don't either. Call us to order. ©1995 Bare Bones Software, Inc. P.O. Box 108 Bedford, MA 01730 voice: 508.651.3561 fax: 508.651.7584
Our lawyers insist we say: BBEdit, PopupFuncs and our spiffy logo are trademarks of Bare Bones Software, Inc. Mac and the Mac OS logo are trademarks of Apple Computer, Inc., used under license. PowerPC is a trademark of International Business Machines Corporation. All other trademarks and registered trademarks are properties of their respective holders.
BBEdit was produced without hair spray, aspartame, or artifical colors, and only a few gerbils were inconvenienced during testing and production.

Power Mac and Macintosh Developers:

FIND OUT FAST WHAT'S GOING ON IN MEMORY

THE MEMORY MINE™

- See memory allocation in any open heap at a glance.
- Easily spot memory leaks.
- Flags heap corruption when it happens.
- Works with source level debugger to let you find memory problems fast.
- Stress applications on the fly with Purge, Compact, and Zap.
- Allocate memory at will for precise stress testing.
- Log heap data - easily document heap status over time.
- No need for source code: nothing inserted in code; no patches to the system.
- Works with 24-bit, 32-bit, and modern memory managers.

For Macintoshes with 68020 or better. Requires System 7.0 or later.

only $99 US
Order now from Adianta, Inc.
Phone: (415)781-8052 • FAX: (415)781-8053
AppleLink:ADIANTA • AOL:Adianta • Internet:adianta@aol.com

For VISA, MC, or American Express orders by mail, fax, or Applelink, please include name, address, card number, expiration date, and phone number or email address.

Also available through the MacTech Mail Order Store and APDA

for more information contact

Adianta, Inc. • 582 Market St #911 • San Francisco, CA 94104

```
    ADDQ.W    #1, D1
    CMPI.W    #NumSize, D1          ; If top number zero,
    BEQ       @5                    ; go to end
    BRA       @3                    ; Loop
@4  SUBQ.W    #1, D1
    BPL       @X
    CLR.W     D1
@X  MOVEQ     #NumSize - 1, D0      ; D0 holds position
                                    ; in top number
    MOVE.L    src2, D7              ; Bottom number
                                    ; is one longword
    CLR.L     D2                    ; D2 is carry register
@1  MOVE.L    00(A0, D0.W*4), D4  ; Get longword of
                                    ; top number
    MULU.L    D7, D5:D4             ; Do 64-bit multiply
                                    ; by bottom number
    ADD.L     D2, D4                ; Add carry
    CLR.L     D2                    ; Use D2 as dummy to
                                    ; extend carry
    ADDX.L    D2, D5                ; Add zero with carry
    MOVE.L    D5, D2                ; High longword
                                    ; becomes carry
    MOVE.L    D4, 00(A1, D0.W*4)  ; Put partial product
                                    ; into result
    SUBQ.W    #1, D0                ; Loop through all
    CMP.W     D1, D0                ; longwords in top #
    BGE       @1
@5  LEA       pro, A0               ; Copy product to "dst"
    MOVEA.L   dst, A1
    MOVEQ     #NumSize/4-1, D0
@2  MOVE.L    (A0)+, (A1)+
    MOVE.L    (A0)+, (A1)+
    MOVE.L    (A0)+, (A1)+
    MOVE.L    (A0)+, (A1)+
    DBF       D0, @2
    MOVEM.L   (A7)+, D0-D7/A0/A1
    }
}
```

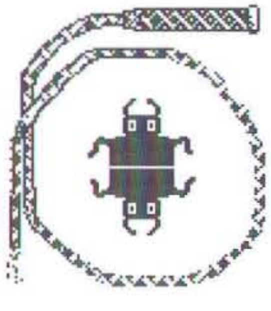
Divide

```
/* Divide end (dividend) by sor (divisor) and put quotient in dst. Remainder will end
   up in end */

void Divide(BigFixed *end, BigFixed *sor, BigFixed *dst)
{
    long      pq;
    BigFixed  quo, line;

    asm
    {
    MOVEM.L   D0-D7/A0-A4, -(A7)
    LEA       quo, A0               ; Clear quotient
    MOVEQ     #NumSize/4-1, D0
@Z  CLR.L     (A0)+
    CLR.L     (A0)+
    CLR.L     (A0)+
    CLR.L     (A0)+
    DBF       D0, @Z
    MOVEA.L   end, A0               ; Move dividend's
                                    ; address into A0
    CLR.W     D0                    ; D0 contains current
                                    ; position in dividend
    MOVEA.L   sor, A1               ; Move divisor's addr
    CLR.W     D1                    ; into A1
@2  TST.L     00(A1, D1.W*4)        ; This loop finds the
                                    ; first non-zero
    BNE       @G                    ; longword of the divisor
    ADDQ.W    #1, D1                ; & stores the pos in D1
    BRA       @2                    ; Loop until something
                                    ; is found
@G  TST.L     00(A0, D0.W*4)        ; Find first longword
    BNE       @1                    ; of dividend
    ADDQ.W    #1, D0
    CMPI.W    #NumSize, D0          ; If whole dividend zero,
    BEQ       @H                    ; go to end
    BRA       @G                    ; Loop
@1  LEA       quo, A2               ; Address of quotient
                                    ; is stored in A2
    MOVEQ     #NumSize, D2          ; D2 contains current
                                    ; position in quotient
    SUB.W     D1, D2                ; First position will
    SUBQ.W    #1, D0                ; be NumSize - D1
    CMP.W     D0, D1                ; If dividend smaller
                                    ; than divisor,
    BLE       @H                    ; then go to end
    ADD.W     D0, D2                ; Add to quotient pos
@C  CLR.L     pq;                   ; "pq" holds partial
                                    ; quotients
@9  MOVE.L    00(A0, D0.W*4), D3  ; Take a quadword from
                                    ; dividend to use
    MOVE.L    04(A0, D0.W*4), D4  ; as partial dividend
    MOVE.L    00(A1, D1.W*4), D5  ; Take a longword from
                                    ; divisor and
    ADDQ.L    #1, D5                ; add one
    BNE       @E                    ; If divisor not zero,
                                    ; go do division
    MOVE.L    D3, D4                ; Else, we are dividing
                                    ; by $10000, so move
    BRA       @F                    ; high longword of
                                    ; dividend into quotient
@E  DIVU.L    D5, D3:D4             ; Do 64-bit division
@F  BEQ       @D                    ;  Branch if quotient zero
    ADD.L     D4, pq                ; Add quotient to
                                    ; partial quotient
    PEA       line                  ; Multiply quotient
                                    ; by divisor
    MOVE.L    D4, -(A7)             ; and store in "line"
    MOVE.L    sor, -(A7)
    JSR       MultQ
    ADDA.W    #12, A7
    LEA       line, A3              ; Get quotient-divisor
                                    ; product
    ADDA.W    #NumSize * 4, A3      ; Now subtract
                                    ; from dividend
    MOVEA.L   A0, A4                ; Cpy dividend addr to A4
    MOVE.L    D2, D5                ; D2 holds lowest pos
    LSL.W     #2, D5                ; of dividend to sub from
    ADDA.W    D5, A4                ; Add it to partial
    MOVEQ     #NumSize, D7          ; dividend address
```

Gives you the Information to *Program your Best!*

The Debugger V2 & MacNosy

by Steve Jasik



Information

Control

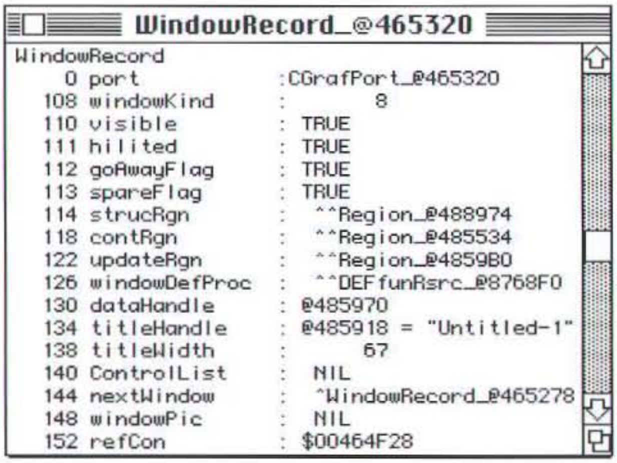
The Debugger is a low and high-level symbolic Debugger that runs in a full multi-window Macintosh environment. You can trace program execution, view the values of variables, etc. ***of both 68K and PowerPC programs***.

MacNosy is a global interactive disassembler that enables one to recover the source code of any Mac application, resource file or the ROM.

When you compare features of the different debuggers, note that *only one* has all the below features to help you get your job done, and *only one* has MacNosy to help you debug any program in a full system (6.0x or System 7.x) environment symbolically!

It is the *only* debugger to use the MMU to protect your CODE resources and the rest of the system from the program you are debugging. With MMU Protection you can find errors when they happen, not millions of instructions later! (Macintoshs with 68030 CPUs only).

The Debugger is the debugger of choice at: Adobe, Aldus, Claris, Electronic Arts, Kodak, Metrowerks, etc.



```
WindowRecord_@465320
WindowRecord
    0 port              :CGrafPort_@465320
  108 windowKind        :        8
  110 visible           :  TRUE
  111 hilited           :  TRUE
  112 goAwayFlag        :  TRUE
  113 spareFlag         :  TRUE
  114 strucRgn          :   ^^Region_@488974
  118 contRgn           :   ^^Region_@485534
  122 updateRgn         :   ^^Region_@4859B0
  126 windowDefProc     :   ^^DEFfunRsrc_@8768F0
  130 dataHandle        :  @485970
  134 titleHandle       :  @485918 = "Untitled-1"
  138 titleWidth        :       67
  140 ControlList       :   NIL
  144 nextWindow        :   ^WindowRecord_@465278
  148 windowPic         :   NIL
  152 refCon            :  $00464F28
```

An example of a structured data display window

Its Features Include:

- **Symbolic Debugging** of *any* Macintosh program, ROM, or *code resource* (DRVRs, XCMDs, INITs, PDEFs, 4DEXs ..)
- **Source level debugging for Metrowerks & MPW** compiled programs (C++, C, Pascal, Fortran, ...), and an Incremental Build System with instant Link for superfast development.
- **Object Inspector for MacApp 3 programs**
- **Source level debugging of Think C™ projects**
- **Includes a program (CoverTest) to interactively do Code Coverage analysis for SQA testing, etc.**
- **Simultaneous Symbolic debugging of multiple "tasks"**
- **Fast Software Watchpoint** command to find clobbered variables
- **Sophisticated error check algorithms** such as Trap Discipline (Argument Checking), Handle Zapping, Heap Scramble and Heap Check to detect program errors before they become disasters
- **Structured display** of data (hypertext) with user definable structures while debugging
- **Conditional breakpoints** to help filter out redundant information
- **Continuous Animated Step Mode** to watch your program execute instruction by instruction
- **Detailed symbolic disassembly for both 680x0 and PowerPC** with symbol names, labels, cross ref maps, – make it possible to ferret out the secrets of the ROM, etc.
- **"Training Wheels"** for the PowerPC disassembler to help you learn the opcodes

The Debugger V2 & MacNosy: $350

Runs on all Macs. Call For Group prices or Updates. Visa/MC Accepted.

Available from: Jasik, APDA, Frameworks or ComputerWare (800-326-0092).

Jasik Designs • **343 Trenton Way, Menlo Park, California 94025** • **(415) 322-1386**
Internet: macnosy@jasik.com • **Applelink: D1037**

Superior Copy Protection

- ✶ MAC*ll*ock makes you money
- ✶ MAC*ll*ock is SAFE, TRANSPARENT & EASY
- ✶ MAC*ll*ock is aggressively priced
- ✶ MAC*ll*ock works on Macs & Power PC's
- ✶ NET*ll*ock provides LAN license control
- ✶ Broad Product Range includes PARA*ll*ock (PC)

Spending $25+ for copy protection should get you more than a pretty dongle. Please let us explain why MAC*ll*ock's micro processor is superior to old fashioned ASIC technology. Call us TODAY and order your Developer API. We will also send a kit to get you started on filing your copyright application.

EAGLE DATA PROTECTION

800-909-3141

On CompuServe GO PIRACY

350 S 400 East, Ste 101 ● Salt Lake City, Utah 84111 ● Tel 801-363-7300 Fax +801-538-0200

Call TODAY & Receive a FREE Copyright Filing Kit

```
    SUB.W    D1, D7              ; D7 holds number of
    SUBQ.W   #1, D7              ; longwords to subtract
    SMI      D5
    MOVE.L   (A4), D6            ; Subtract least
    SUB.L    -(A3), D6           ; significant longwords
    MOVE.L   D6, (A4)
    TST.B    D5                  ; If there are not any
                                 ; more longwords
    BNE      @B                  ; to subtract, branch
@A  SUBX.L   -(A3), -(A4)        ; Subtract rest of
    DBF      D7, @A              ; longwords with borrow
@B  BRA      @9                  ; Loop and divide again
@D  MOVE.L   pq, D4              ; Get accumulated
                                 ; partial quotient
@8  MOVE.L   D0, D5              ; Now check to see
    ADDQ.L   #1, D5              ; if partial dividend is
    MOVE.L   D1, D6              ; less than divisor
@5  MOVE.L   00(A1, D6.W*4), D7 ; Get longword of divisor
    CMP.L    00(A0, D5.W*4), D7 ; Compare with longword
                                 ; of dividend, branch
    BCS      @3                  ; if dividend bigger
    BNE      @4                  ; If divisor bigger,
                                 ; then branch
    ADDQ.L   #1, D5              ; If these two longwords
    ADDQ.L   #1, D6              ; were =, then compare
    CMP.W    D2, D5              ; next two longwords and
    BLE      @5                  ; continue
@3  MOVEA.L  A1, A3              ; Subtract divisor from
    ADDA.W   #NumSize * 4, A3    ; partial dividend
    MOVEA.L   A0, A4
    MOVE.L   D2, D5
    LSL.W    #2, D5
    MOVEQ    #NumSize, D7
    ADDA.W   D5, A4
    SUB.W    D1, D7
    SUBQ.W   #2, D7
    SMI      D5
    MOVE.L   (A4), D6
    SUB.L    -(A3), D6
    MOVE.L   D6, (A4)
    TST.B    D5
    BNE      @6
@7  SUBX.L   -(A3), -(A4)
    DBF      D7, @7
@6  ADDQ.L   #1, D4              ; Add one to quotient
    BRA      @8                  ; Loop and compare again
@4  MOVE.L   D4, 00(A2, D2.W*4) ; Store partial quotient
                                 ; in whole quotient
    ADDQ.W   #1, D2              ; Increase quotient pos
    ADDQ.W   #1, D0              ; Increase dividend pos
    CMPI.W   #NumSize, D2
    BLT      @C                  ; Loop until done
@H  LEA      quo, A0             ; Copy quotient to "dst"
    MOVEA.L  dst, A1             ; Remainder will automat-
    MOVEQ    #NumSize/4-1, D0    ; ically end up
@0  MOVE.L   (A0)+, (A1)+        ; in dividend
    MOVE.L   (A0)+, (A1)+
    MOVE.L   (A0)+, (A1)+
    MOVE.L   (A0)+, (A1)+
    DBF      D0, @0
    MOVEM.L  (A7)+, D0-D7/A0-A4
    }
}
```

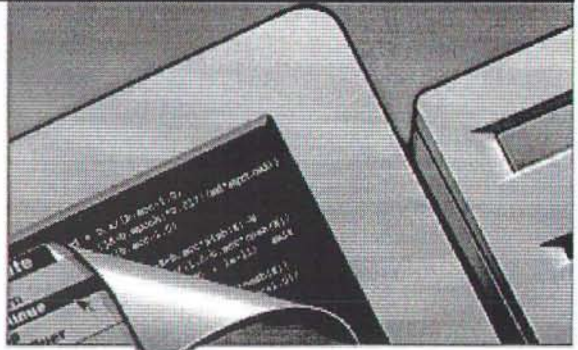
Compare

```
/* Compare src1 and src2.  Returns 1 if src1 > src2, 0 if they're equal, and -1 if src1 <
  rc2. */

short Compare(BigFixed *src1, BigFixed *src2)
{
    asm
    {
    MOVEM.L  D1/D2/A0/A1, -(A7)
    MOVEA.L  src1, A0            ; Get src1's address
    MOVEA.L  src2, A1            ; Get src2's address
    MOVEQ    #1, D0              ; Start with +1
    MOVE.L   (A0)+, D2
    CMP.L    (A1)+, D2           ; Compare 1st longwords
    BLT      @1                  ; If src1 less, branch
    BNE      @2                  ; If !=, src1 must
    MOVE.L   (A0)+, D2           ; be greater
    CMP.L    (A1)+, D2           ; Cmp 3 more longwords
    BCS      @1                  ; (Unsigned)
    BNE      @2
    MOVE.L   (A0)+, D2
    CMP.L    (A1)+, D2
    BCS      @1
    BNE      @2
    MOVE.L   (A0)+, D2
    CMP.L    (A1)+, D2
    BCS      @1
    BNE      @2
    MOVEQ    #NumSize/4-2, D1    ; Number of longwords
                                 ; remaining / 4
@3  MOVE.L   (A0)+, D2           ; Compare 4 longwords
    CMP.L    (A1)+, D2
    BCS      @1
    BNE      @2
    MOVE.L   (A0)+, D2
    CMP.L    (A1)+, D2
    BCS      @1
    BNE      @2
    MOVE.L   (A0)+, D2
    CMP.L    (A1)+, D2
    BCS      @1
    BNE      @2
    MOVE.L   (A0)+, D2
    CMP.L    (A1)+, D2
    BCS      @1
    BNE      @2
    DBF      D1, @3              ; Loop
    CLR.L    D0                  ; If we get here,
    BRA      @2                  ; then src1 = src2
@1  MOVEQ    #-1, D0             ; src1 is less
@2  MOVEM.L  (A7)+, D1/D2/A0/A1
    }
}
```

World's Leading FORTRAN 77 for Macintosh

68K and Power Macintosh versions

- *full ANSI 77 native compiler*
- *faster execution speed*
- *graphical source-level debugging*
- *two complete graphics packages*
- *make utility, MRWE application framework*
- *System 7.5 compatible, MPW included*
- *Windows 95/NT version also available*



absoft
development tools and languages

Visit us now at http://www.absoft.com

2781 Bond Street Rochester Hills MI 48309 • (810) 853-0050 • Fax (810) 853-0108 • sales@absoft.com

GET YOUR NAME "IN LIGHTS"

Have you ever thought about writing an article? The editorial staff at MacTech Magazine would like to personally invite you to write for the publication. While writing an article is not difficult, it does take a bit of time. But, you get to share your knowledge with the community, see your names "in lights", and ... you get paid for your efforts!

To get started, you can download the MacTech Writer's Kit from one of our online support areas. Here you will find information on how to submit an article – and it comes with examples, templates and style sheets. Feel free to e-mail us with questions.

CALL FOR ARTICLES

Many of you have asked "what topics would we like to see?" Our Editor Board and Staff has provided insight as to what the magazine will cover in 1996. Our goal at the magazine is to publish articles on diverse topics that are sure to keep our readers both interested and well-informed throughout the year. If you'd like to be part of this group, think about what you know about – that's what we'll be most interested in – and what you'll be the best at writing. As a guideline, you can think about one of the following issues.

For example, topics that include articles that teach debugging techniques and advocate good debugging tools. This continues our theme about making software more reliable. We will continue to talk about "cool" Apple technology such as QuickTime VR, QuickDraw GX, Dynamic Languages; or as our Editor-at-Large Scott T Boyd said, "software that takes us out of the Stone Age."

There will be tips on how to get your business on the Internet and how business models are shaping up. More heavy-hitting deep technology articles will be included, such as the recent articles on CGI development, C++ Exception Handling, etc. We'll see continued coverage on the developments in OpenDoc, as those in the industry decide what to make of this new software model.

MacTCP is another area of importance, as more people get excited about providing Macintosh-quality software for Internet users. You will see additional coverage on Visual Programming. You should also expect to see more on cross platform development. And, even though it's a moving target, we will be covering Apple's next System Software release – Copland.

For Macintosh Programmers & Developers
MacTech™
MAGAZINE

E-mail us with your idea and let's talk. You never know, you might get your name "in lights"!

Client/Server Object Database Engine

neoShare™

Collaborative Computing

NeoShare™ is a collaborative computing engine that object-oriented developers use to create data intensive distributed applications. Its client/server architecture provides shared access to objects by multiple clients. The server can co-exist within a client application or operate in the context of a dedicated process. Clients can reside on the same machine as the server or on other machines across the network. No other multi-user object database engine can match NeoShare's feature set, impressive performance, incredibly small resource footprint, easy to use yet extensible programming interface and generous pricing structure.

Full Featured

NeoShare consists of a set of C++ classes that further extends the features of NeoAccess™, our cross-platform object database engine. Taken together, NeoAccess and NeoShare form a feature-rich groupware toolkit that addresses a complete set of collaborative development requirements, including; object persistence, organizing, searching, transporting, sharing, locking, updating and concurrency control.

The collaborative capabilities of NeoShare include; client/server connectivity, concurrency control, object locking, transactions, a hot-link facility, client-side caching to reduce network traffic and object overwrite detection. These two professional quality development tools form the foundation upon which data intensive distributed applications are built.

Full Source Code

NeoLogic has taken a frameworks approach with our object-oriented development tools. In much the same way that application frameworks are used to construct the front-end of an application, NeoAccess and NeoShare are the frameworks you use to build your application's collaborative back-end capabilities. This allows you to focus on the value-added portion of the application - the part that you know best.

As is the case with virtually all object frameworks, the NeoShare Developer's Bundle comes complete with full source code. So you have the security of knowing that NeoShare technology is always at your fingertips. And our extensible architecture allows you to evolve your toolkit into domain-specific areas.

Easy to Use

Migrating your application source code from NeoAccess to NeoShare is easy because the programming interface of both products is virtually identical. This interface is designed around the concept of minimum visible complexity. NeoShare naturally extends NeoAccess to add object sharing between clients and server. Objects obtained from a server across the network by a client are real C++ objects. You treat these objects as you would any other object.

NeoShare includes two concrete subclasses of the NeoAccess database class; a client and a server. Virtually all the complexities associated with sharing objects across a network have been hidden behind the interface to these database subclasses. In application code, a server database object appears very much like a local database, except a server database also handles requests received from client applications. As clients request objects from a server, the server maintains a set of locks associated with those objects. When the state of an object in a server database changes, all clients holding locks on the object are automatically sent the object's new state.

A client object also supports the same application programming interface as a local database. Yet the implementation of this interface may involve sending a request to the remote server. The server performs the requested operation and sends the results of this operation back to the client. The caller of the client database needn't be aware of the fact that the database object on which it relies is in fact delegating operations to a server database across a network interface. The network is transparent to the application developer.

Affordable

The NeoShare Developer's Bundle, which includes NeoAccess, sells for just $1299 per developer with no runtime licensing fees. It includes full source code, sample applications, online documentation, and 30 days of online technical support. Site licensing, as well as educational and volume discounts are also available.

Now on the CodeWarrior CD!
See the NeoLogic folder on the root
of the CD-ROM for details.

No Runtime Licensing Fees!
Use NeoAccess and NeoShare in all
your development for one low price!

neo•logic

Professional quality object-oriented development tools

1450 Fourth St. • Suite 12 • Berkeley • CA • 94710 • (510) 524-5897 • neologic@neologic.com

SYMANTEC TOP 10

By Levi Brown, Symantec Corp.

SYMANTEC.™

This monthly column, written by Symantec's Technical Support Engineers, aims to provide you with technical information based on the use of Symantec products.

Q: I set my project's preferred and minimum memory partition sizes in the "Project Type" window, but when I build my application, the preferred and minimum memory partition sizes are always set to a different value. What is wrong?

A: The project's resource file contains a `SIZE` resource that always overrides the value set under the **Project**, **Options...**, "Project Type" window. To solve this problem, open the `ProjectName.rsrc` resource file with ResEdit and delete the `SIZE` resource.

Q: I have heard of this program called Discipline that works with MacsBug to provide a higher level of error checking. What does it do, and does it work with the Symantec Debugger?

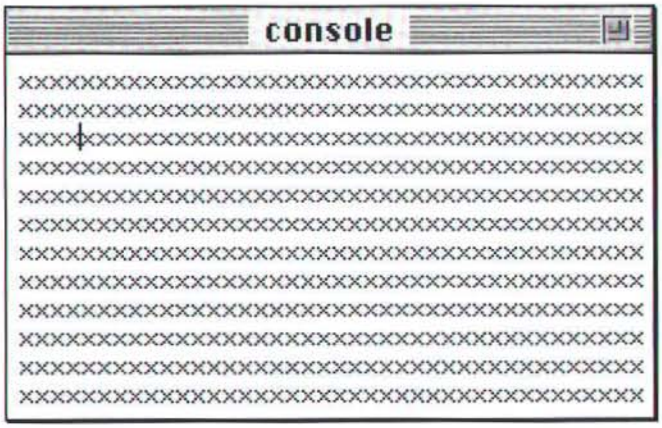
A: Discipline is a system extension that works with MacsBug to check the arguments passed to Toolbox routines. Discipline is built into MacsBug 6.5d12. To access it, you use the dcmd `dizy`.

Our debugger has not been tested thoroughly with Discipline. Therefore, it is possible that you may run into some incompatibilities. Discipline has a habit of revealing problems with many different programs including the Finder, which makes it difficult to use as a debugging tool.

Q: In UNIX there is a collection of routines, declared in `curses.h` that can be used to manipulate the cursor in a console-like display. In my Symantec C++ 8.0 project using the console window how can I achieve similar results?

A: Most of the functionality of the console window is declared in `console.h`. Here is an example of how to move the cursor to a specific point, and clear everything in the window from that point down.

```
#include <stdio.h>
#include <console.h>

main()
{
   int i, ch, rows = 12, cols = 40;

   console_options.nrows = rows;
   console_options.ncols = cols;
   cshow( stdout );
   csetmode( C_NOECHO, stdout );
                        // Set mode to C_NOECHO, so
                        // getchar doesn't move the
                        // cursor.
   for( i=0; i<rows*cols; i++ )
     putchar( 'x' );
   cgotoxy( 5, 3, stdout );
   ch = getchar();
   ccleos( stdout );
}
```



```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Before the call to `ccleos()`

You've spent **enough** time in development.

Don't let your installer keep you from **go**ing **golden**.

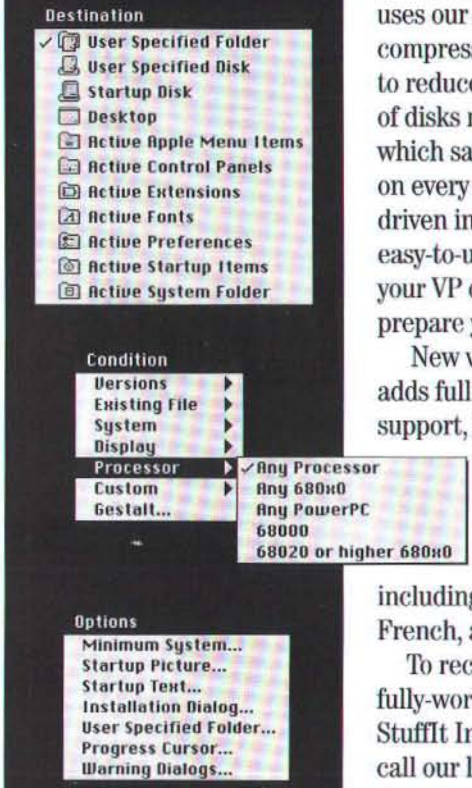
Just because your software is done doesn't mean your work is. You still have to write an installer. And that can add precious days or even weeks.

With Aladdin Systems' proven installation standard, StuffIt InstallerMaker,™ you can have your PowerMac® or 680x0 product ready to ship literally within minutes, and save money to boot.

StuffIt InstallerMaker uses our advanced compression technology to reduce the number of disks needed to ship, which saves you money on every unit. Its menu-driven interface is so easy-to-use that even your VP of Sales could prepare your installer.

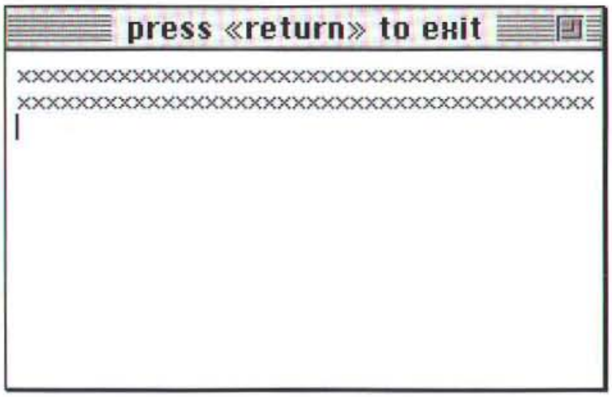
New version 2.0.2 adds full PowerMac support, improved scriptability, and expanded localization, including German, French, and Japanese.

To receive a free, fully-working copy of StuffIt InstallerMaker, call our licensing department today at (408) 761-6200.



STUFFIT INSTALLERMAKER

© 1994 Aladdin Systems, Inc. 165 Westridge Drive, Watsonville, CA 95076. Fax: (408) 761 6206. America Online, AppleLink: ALADDIN. Internet: aladdin@well.com. InstallerMaker is a trademark of Aladdin Systems, Inc. All other products are trademarks of their respective holders.



```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

After the call to `ccleos()`

Q: I'm using the ToolBox call `GetPicture()` and I want to clean up the memory after I am done with the picture. How do I do that?

A: Pass the handle that `GetPicture()` returns to `ReleaseResource()`. Do not call `DisposeHandle()` on a handle to a resource or you may corrupt the resource map.

Q: When I try to use the QuickDraw global variables in my Symantec C++ 8.0 project I get undefined symbol errors. I've used the same globals many times before using 7.0 and it worked just fine. What is happening here?

A: With the introduction of Apple's Universal Headers, access to the QuickDraw globals changed. Your 7.0 project was probably referring to the QuickDraw globals in the "old" style and using the original header files, not the Universal Headers. You will need to access the QuickDraw globals through the qd struct, defined in `Quickdraw.h`. i.e.:

```
CopyRgn(thePort->visRgn, thePort->clipRgn);
```

would become

```
CopyRgn(qd.thePort->visRgn, qd.thePort->clipRgn);
```

Q: I have a large 8.0 project with many files and I frequently use the **Find...** command under the **Search** menu. Every time I use it I have to set up the find parameters each time. Is there a way to set default search parameters?

A: Yes! In your Symantec C++ folder there is a folder called `(Scripts)`, inside which you will find a compiled script called `Startup`:

```
tell application "Symantec Project Manager"
   -- Set the default search state.
  set user search state to ¬
    {wrap around:true, ¬
     ignore case:true, ¬
     files to search:only sources, ¬
     exclude system:true, ¬
     exclude precompiled header:true}
end tell
```

You can set the default parameters for the find in the indexed list shown above.

Q: I get link errors with a class that has static data members. Why is this?

A: If you declare a class with static data members you need to initialize those data members in the global data space outside the class, so that memory is allocated for them. If you do not initialize the static variables in global data space outside the class then the compiler does not have a location for the static data members in memory. Thus, you get a link error when you attempt to access that variable. Example:

```
class A
{
public:
   static short x;
   static char  y;

   char z;
};

short A::x = 0;         // Note: These variables are initialized
char  A::y = 'a';       // outside the class declaration in
                        // order to allocate memory for them.
```

Q: The following code used to work (up through 8.0), but in 8.0.1 it doesn't. Why?

```
class A
{
   friend class B;
public:
   A();
   ~A();
   B *GetFriend(int index);
};

class B
{
public:
   B();
   ~B();
   int index;
};
```

I get an error on line 7 "semicolon expected". If I put `friend` before `B *GetFriend(int index)`, it works. Is it a compiler bug, or is it a new ANSI rule of some kind that I didn't know about?

A: A couple things are happening here:

First, the code above compiled in previous versions due to a compiler bug. 8.0.1 corrects this error. Basically, the syntax is wrong. The class `A` has no idea what `B` is yet, and thus the compiler complains. If you put a forward declaration, `class B;`, in front of the declaration of A this will compile fine.

Second, the statement `friend B *GetFriend(int index)` tells the compiler that `GetFriend` is a friend function, and therefore has access to A's protected and

ANNOUNCING

LS OBJECT PASCAL V1.0

OPTIMIZING COMPILER FOR 68K AND POWERMAC

COMPLETE PROGRAMMING ENVIRONMENT WITH OVER 100MB OF TOOLS AND DOCUMENTATION

SOURCE LEVEL DEBUGGERS

VISUAL INTERFACE GENERATOR

SOURCE CODE GENERATOR

PLUS

NATIVE PORTING KIT FOR MACAPP 2.0!

COMING SOON! SUPPORT FOR SYMANTEC PROJECT MANAGER 8.0!

CALL OR WRITE FOR MORE INFO:

1-800-252-6479
1-703-689-9593 fax
Language Systems Corp.
100 Carpenter Dr. Sterling VA 20164
http://www.langsys.com/langsys

MacForth Plus & Power MacForth

The Language of Innovation is now available native on the Power Macintosh.

FEATURES

- Royalty Free Turnkey Compiler
- Text Editor
- Online Glossary Tool
- System 7 Compatibility
- 68020 Assembler
- 68881/2 Co-processor Support
- High Level Graphics
- Toolbox Support
- Extensive Documentation
- Reasonable Upgrades
- Optional Tools Disks
- MacForth Plus $199
- Power MacForth $199

Creative Solutions, Inc.

4701 Randolph Road, Suite 12
Rockville, MD 20852
301-984-0262 or 1-800-FORTH-OK *(orders)*
Fax: 301-770-1675 Applelink: CSI

private members. The statement `B *GetFriend(int index)` tells the compiler that `GetFriend` is a member function of A. In both cases the `friend class B` does nothing more than allow all members of B access to A's private and protected members.

Q: I have made a floating window view in Visual Architect, but when I go to attach it to a command, it does not appear in the list of views to open. How do I open it?

A: When you make a floating window view, VA derives a class from `CFloatDirector`; we will call it `CMyFloatingWindow`. You need to call `OpenWind()` to make the window appear (`ShowWind()` only undoes a `HideWind()`). The easiest way is to call it from within `CMyFloatingWindow::MakeNewWindow()`, right after the object is created; the window will appear immediately upon launch. To have it open from a command (i.e. menu or button), create the command by choosing **Commands...** from the **Edit** menu in VA. Enter the name for your command (e.g. `cmdOpenMyFWindow`), choose **CApp** from the **In Class:** popup menu and choose **Call** from the **Do:** popup menu.

Next, attach the command to the menu or button. If attaching to a menu: choose **Menus...** from the **Edit** menu, select the menu, click on **Edit Menu Items**, select the item and choose the command you created from the **Command:** popup menu. If attaching to a button: select the button in VA, get **Pane Info** from the **Pane** menu, open the CButton triangle and choose the command you created from the **Command:** popup menu.

Visual Architect will generate an empty function, `DocmdOpenMyFWindow` inside `x_CApp.cp`. Call `OpenWind()` from here.

Note: Calling from outside the `CMyFloatingWindow` requires that you have a pointer to the `CMyFloatingWindow` object. You can find this declared as a global in `x_CApp.cp` in the form:

```
extern CAFloatingWindow *gCAFloatingWindow;
```

If you wish to make the call to `OpenWind()` from someplace other than `x_CApp.cp`, you must redeclare it (as above) and `#include CMyFloatingWindow.h` in the file where you want to use it.

Q: When I try to run the Symantec Debugger it quits with an error that says, "DebugServices cannot set trace bit." How do I set the trace bit? I can't find any options to set a trace bit in any of the project option debugging screens.

A: Most likely, you are running a Pre-System 7.5 operating system. You will need to put the items that are in the Pre-System 7.5 Additions folder into your system folder. The Pre-System 7.5 Additions folder is found in the Apple Software folder on your Symantec C++ 8.0 CD-ROM.

BONUS:

Q: I'm porting my code from 7.0.x to 8.0 and I get an "undefined symbol `MBarHeight`" when I try to compile. Why do I get a linker error?

A: With the introduction of Apple's Universal Headers you can no longer access the variable `MBarHeight` directly. You will need to include `LowMem.h` and use `LMGetMBarHeight()`, and `LMSetMBarHeight()`, to access menu bar height. Access to other low memory variables has been changed in similar ways, so it's a good idea to look in `LowMem.h` if you plan on using low memory variables.

SPECIAL THANKS TO:

Glenn Austin, Mark Baldwin, Craig Conner, Rick Hartmann, Michael Hopkins, Steve Howard, Noah Lieberman, Andy McFarland, Chris Prinos, and Kevin Quah.

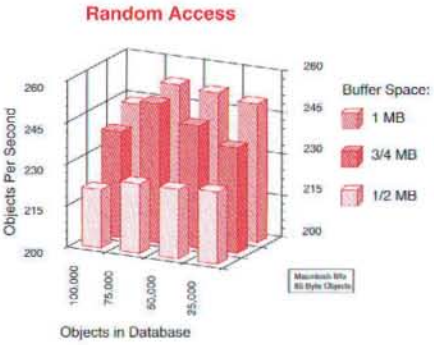
neoAccess™

Cross-Platform Object Database Engine

Full Featured

Now you can embed the power of a full-featured object-oriented database engine into your **Macintosh**, **Windows**, **Unix** and **DOS** based applications. No other object database engine can match NeoAccess's impressive feature set, including: Blobs, part lists, iterators, swizzlers, temporary objects, multiple indices on a class, inverted indices, stored queries, schema evolution, a powerful relational query mechanism, a streams-based i/o model and incredible performance.

High Performance



Internally, NeoAccess uses extended binary trees and binary search algorithms to achieve optimally short access times. Its automatic query optimizer ensures that queries always use the fastest access path to objects. And indices are dynamically combined, collapsed and compressed to keep access times to an absolute minimum as the contents of a database changes. NeoAccess's object caching boosts performance by keeping objects in memory even after being disposed of by the application. Your application's memory size is reduced because only those objects of immediate interest need to be in memory at any one time, not the entire file.

Full Source Code

We've taken a frameworks approach toward object persistence and database technology. In much the same way that application frameworks are used to construct the front-end of an application, NeoAccess is the framework you use to build your application's back-end. As is the case with virtually all object frameworks, the NeoAccess Developer's Toolkit comes complete with ***full source code***, for all major application frameworks including Metrowerks' **PowerPlant**, Symantec's **THINK Class Library** and Apple's **MacApp** on the Macintosh and **Microsoft's Foundation Classes**, Inmark's **zApp** and Borland's **ObjectWindows** in Intel-based environments. It can even be used without a framework or in one that you've designed.

Easy to Use

The programming interface is designed around the concept of minimum visible complexity. Application-specific objects inherit persistence properties from a NeoAccess base class. These objects are organized in the database primarily by class. But NeoAccess also knows how classes are related. So multiple classes can be searched in a single operation. And of course objects of any particular class can be organized using multiple indices. NeoAccess is unique in that it allows objects to be located based on abstract selection criteria or based on their relationship to other objects. There's literally no database administation to deal with – NeoAccess takes care of all the details. NeoAccess also includes a Blob mechanism which allows free-form variable-length data to be stored in databases with the same ease as fixed-length objects. NeoAccess even includes a powerful set of keyed iterators for traversing indices and part lists. Keyed iterators have the unique ability to iterate over only those objects in a set that match a given selection criteria. Your users will appreciate NeoAccess because databases are completely self-contained in a single document file. So users can treat a database file as they would any other document.

Proven

NeoAccess has been commercially available since 1992. Thousands of commercial and in-house applications based on NeoAccess technology have already been deployed. NeoAccess can help your organization deliver powerful products in a more timely fashion than you ever imagined possible.

Affordable

NeoAccess's best feature is its price. The NeoAccess Developer's Toolkit sells for just $749 per developer with absolutely ***no runtime licensing fees***. It includes ***full source code***, numerous sample applications, 500 pages of documentation, and 30 days of technical support. So what are you waiting for?

Now on the CodeWarrior CD!
See the NeoLogic folder on the root
of the CD-ROM for details.

No Runtime Licensing Fees!
Use NeoAccess and NeoShare in all
your development for one low price!

neo•logic

Professional quality object-oriented development tools

1450 Fourth St. • Suite 12 • Berkeley • CA • 94710 • (510) 524-5897 • neologic@neologic.com

PASCAL ACTIVISTS

By Brian Arnold and Guy McCarthy

MacApp Pascal Rides Again

Volunteers convert MacApp Pascal to generate native PowerPC code

In 1992, Apple stunned Macintosh developers by announcing that it was dropping MPW Pascal support from its compiler suite and from MacApp, their premier application framework. This was a shock, given that Apple had been advising developers for years to program in Object Pascal with MacApp for maximum compatibility with future technology.

MacApp developers split into two camps: those able to make the arduous transition to C++, and those who could not justify the expense or were unwilling to abandon Object Pascal. The latter group faced an uncertain future, and hopes of porting existing MacApp applications to run native on the new platform were slim.

Before the fateful announcement, Apple and Language Systems had been collaborating on the next generation of Pascal compilers for the RISC-based Macintosh. That project, however, was ultimately scrapped when Apple decided to drop multi-language support from its RISC compiler architecture. In December of 1993, development of LS Object Pascal began again, this time using source code from Apple's MPW Pascal compiler as the base. It was a long and difficult task to merge the aging MPW Pascal source code with Language System's FORTRAN technology and update it for the PowerMacintosh.

In May of 1994, the first native Pascal sample program was shown to an enthusiastic crowd of Pascal supporters at the World Wide Developers' Conference. Shortly after, a grass-roots cooperative of Pascal MacApp developers was formed to investigate the possibility of porting MacApp native with the Language Systems compiler. The MacApp2PPC co-op (as it has come to be known) was organized by Brian Arnold of Lumina Decision Systems. This group of volunteers proceeded to accomplish what the corporations could not.

Over the next several months, contributors on three continents began the lengthy process of converting the entire body of MacApp 2.0.1 source code. In October, an FTP site and Web page were operational with a preliminary version of the scripts and documentation (see below). In November, key changes to the compiler's runtime architecture were implemented to satisfy requirements of the group. By December 1994, all of the converted source code was compiling successfully with Language Systems' beta LS Object Pascal compiler. Only runtime issues remained.

In February, the target release date of May 1995 was set to coincide with the Worldwide Developer Conference in San Jose, California. Final technical issues were addressed during a 48 hour "Hack Session" on March 17–19 in nearby Menlo Park. Language Systems sent two compiler engineers with Power Macintosh CPUs to aid the effort.

The Hack Session was a success. Participants (some of them operating remotely) included: Masahiro Abe, Brian Arnold, Colleen Barton, Per Bergland, Greg Branche, Larry Hamel, Steven Hopkins, Eric Jackson, David Johnston, Steve Lavagnino,

Brian Arnold and Guy McCarthy – Brian Arnold, Director of Software Development at Lumina Decision Systems, is an ardent supporter of OpenDoc and MacApp 3.3, but his company badly needed PowerPC native versions of legacy Object Pascal MacApp2 code. Guy McCarthy is President and CEO of Language Systems Corp.

Simply the best GUI Building/Event Managing libraries

Tools Plus™ 2.6

Tools Plus gives you the routines you need to create a professional looking user interface. Then we make it work. It's that simple.

Now for CodeWarrior!

✓ CodeWarrior ✓ C/C++ ✓ PowerPC
✓ Symantec ✓ Pascal ✓ 680x0

- Over 170 high-powered "set and forget" routines that automate and enhance: event handling, windows, the tool bar, floating palettes, cursors, buttons, picture buttons, scroll bars, menus (pull-down, hierarchical and pop-up), list boxes, fields, Edit menu, clipboard, Dynamic Alerts, and more…
- Easy to learn and easy to use
- Substantial code reduction
- Dramatic code simplification
- Significantly less debugging
- System 6 and 7 compatible
- For novice, intermediate and advanced programmers
- Runs fast; needs little memory or disk space
- Safer than toolbox routines
- No royalties

OK

Language
○ C/C++
○ Pascal
● Both

Size: 9 10 ● 12 14 18 24 36

☒ Saves Time
☒ Saves Money
☒ Easy to Use

Water's Edge Software

2441 Lakeshore Road West • Box 70022
Oakville, Ontario • Canada L6L 6M9
Phone: (416) 219-5628
Fax: (905) 847-1638

Internet: WatersEdge@eWorld.com
eWorld: WatersEdge
CIS: 73424,2507

Tools Plus for	
Symantec (THINK) C/C++	$149
THINK Pascal	$149
Symantec C/C++ & Pascal	$199
CodeWarrior Bronze	$199
CodeWarrior Gold	$249

(We accept VISA and American Express. Add $10 for shipping.)
* Call for Academic pricing

Free Evaluation Kit:

Available on **AOL, AppleLink, CompuServe, eWorld** and other BBSs.

Disk also available by mail.

Cheryl Lins, Kurt Schmucker, and Larry Tesler. Although the first native execution of a complete MacApp application was still weeks away, important technical issues, such as exception handling, were resolved. In addition, new technology such as floating palettes and other enhancements were merged into the Pascal framework.

In the weeks following the Hack Session, the co-op mounted an incredible effort to get PowerPC native MacApp 2.0.1 up to beta quality in time for the Developer Conference. Support for the Metrowerks Pascal compiler was added. Several key managers at Apple joined in and arranged a distribution license so that Language Systems and Metrowerks could include the PowerPC native MacApp 2.0.1 on their CDs. (Special thanks to Kurt Schmucker for rallying this effort!) The first developer release of MacApp2PPC shipped on CodeWarrior DR6 in May. An improved version shipped on DR7, but owing to a CD pressing glitch, the project files and documentation are wrong; the correct material is available online (see below).

About a dozen developers have already partially or completely ported their applications to PowerPC, and a few of them have already started shipping their native applications. Brian Arnold ported his own application, Analytica™, in under two weeks, and expects to ship by the end of the year.

It is hoped that by January 1996, two versions will be ready: the final, stable MacApp2PPC, and a developer release of an improved version. *MacTech* hopes to report further details in upcoming issues. In the mean time, developers who have LS Pascal can use the netborne release of MacApp2PPC. For more information see the MacApp2PPC web pages at:

http://www.lumina.com/arnold/MacApp2PPC.html

Or, for anonymous ftp:

ftp://ftp.rahul.net/ftp/pub/arnold/macapp2ppc.

To get on the mailing list, write to:
arnold@lumina.com

To receive information on any products advertised in this issue, send your request via Internet: productinfo@xplain.com

FREE INSTALLER!

To: Mac Developers & Product Managers

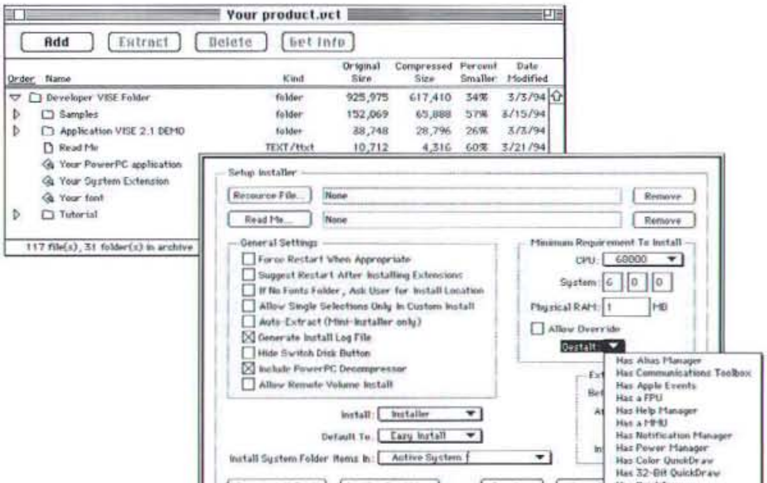
From: MindVision (Creators of Stacker for Macintosh)

Subject: Free Copy of DEVELOPER VISE 3.0

Message: We've got a great new installer for you. Here's your chance to try it for free. No risk, no obligation, no hassle. Call today, don't delay!

Full PowerPC Support

Integrated compression



Order	Name	Kind	Original Size	Compressed Size	Percent Smaller	Date Modified
▽	Developer VISE Folder	folder	925,975	617,410	34%	3/3/94
▷	Samples	folder	152,069	65,888	57%	3/15/94
▷	Application VISE 2.1 DEMO	folder	38,748	28,796	26%	3/3/94
	Read Me	TEXT/ttxt	10,712	4,316	60%	3/21/94
	Your PowerPC application					
	Your System Extension					
	Your font					
▷	Tutorial					

117 file(s), 31 folder(s) in archive

Resource File...: None

Read Me...: None

General Settings

- ☐ Force Restart When Appropriate
- ☐ Suggest Restart After Installing Extensions
- ☐ If No Fonts Folder, Ask User for Install Location
- ☐ Allow Single Selections Only In Custom Install
- ☐ Auto-Extract (Mini-Installer only)
- ☒ Generate Install Log File
- ☐ Hide Switch Disk Button
- ☒ Include PowerPC Decompressor
- ☐ Allow Remote Volume Install

Minimum Requirement To Install

CPU: 68000

System: 6 0 0

Physical RAM: 1 MB

☐ Allow Override

Gestalt:

Install: Installer

Default To: Easy Install

Install System Folder Items In: Active System

No programming required

Fully graphical interface

Call (402) 477-3269

IF YOU PREFER, USE E-MAIL. APPLELINK, AOL: MINDVISION • COMPUSERVE 70253,1437

Join the growing list of companies who use our VISE technology: Adobe, Apple, CE Software, Claris, CompuServe, DeltaPoint, MacroMedia, Radius, Stac Electronics, Symantec, WordPerfect, and many more.

MindVision Software

MindVision Software 840 South 30th St., Suite C, Lincoln, Nebraska 68510
Voice: (402) 477-3269 Fax: (402) 477-1395 AppleLink, AOL: MindVision

VISE COMPRESSION TECHNOLOGY

© 1992-94 MindVision Software. All Rights Reserved. Developer VISE is a trademark of MindVision Software.

CRABB'S APPLE

By Don Crabb

Visually Speaking

It's November, 1995. Do you know where your Mac development efforts are heading? Will they be linked, arm-in-arm, with Metrowerks, Symantec, and Apple, or will you set-off in the direction of one of the Mac's third party niche tools companies, like Main Event, Pictorius, MainStay, QKS, or Digitool? Maybe you're even making the move to client/server development and are hunkering down with Oracle, 4th Dimension, Sybase, or even Filemaker Pro 3.0 (relational) and the FM Pro Server. And how much of your efforts are to develop software for both the Mac and Windows 95?

Visual Basic 4.0 for the Mac?

No matter what direction your Mac development efforts will take you, however, they won't take you to Microsoft's Visual Basic 4.0. MS has said "no way" to porting VB to the Mac, thus adding yet another brick to the wall isolating you from potential customers.

Customers who might buy your Mac products for their Mac ghettos if they could somehow interoperate your Mac apps effectively with their huge installed base of Windows products, most of which are driven by OLE.

Customers you might have reached a little faster if only you'd had the aid of VB 4.0 for the Mac to prototype and develop your Mac apps to live and work in a Windows world.

But Microsoft says "no way." They don't want Mac developers porting their apps to Windows. They want Mac developers to dump the Mac and develop directly for the 32 bit API of Win95 and NT.

The Importance of OLE and VB

No matter how little you think of OLE 2.0, it's not going to go away just because OpenDoc is a better SOM implementation. And Visual Basic 4.0 is the raison d'tre of OLE 2.0. VB 4.0 lives and breathes OLE. It can create every kind of OLE server and container, including in-process servers.

With VB 4.0, the OCX standard has arrived with a flourish. VBX controls are automatically converted to OCX controls. The OCX language dialect is now VB and the runtime engine is shared by all MS Office 95 applications that use VB. And every third party addon module to Office 95.

You cannot, however, create OCX controls in VB, but you'd buy Borland's Delphi 95 for that (yet another top Windows development tool that will not make it to the Mac).

VB 4.0 is also the defacto scripting language for Windows 95, even if MS has not pushed it as such (nor bundled it with the Win95 package as they should). As much as I love AppleScript and scripting on the Mac, VB 4.0 can do things that you can't do nearly as well with AppleScript (even with 3rd party aids like Scripter and FaceSpan), like creating commercial applications or OLE objects. Visual Basic 4.0, in short, is a big deal. And it's a bad deal for Mac developers (and customers) not to have it available for Macintosh.

What's Hot About VB 4.0?

VB 4.0 is what I call a bridge language. It can be used by customers used to write application or system macros and scripts. It can be used by managers to create quickie apps.

About Don Crabb – Don is a contributing editor and columnist for MacTech, MacWEEK, MacUSER, Mac/Chicago, MacToday, Win95User, ComputerUser, The Chicago Sun-Times Features Syndicate, The Springfield Union-News, PC Magazine, and about a million other publications. Don welcomes comments at his Internet address: decc@cs.uchicago.edu. You can also check out his WWW Home page at http://www.cs.uchicago.edu/~decc/.

From idea to delivery...

Complete Software Duplication Service

800-222-6032

FAST

Software Duplication ◆ CD ROM
Labeling ◆ Assembly
Printing ◆ Packaging ◆ Shipping

all handled with top

QUALITY ... SERVICE ... SPEED

Authorized Apple Duplicator
For 11 Years!

International Datawares, Inc
2278 Trade Zone Blvd
San Jose • CA 95131
(408)262-6660
Fax: (408)262-8906
Internet: idikds@aol.com

And it can be used by hard core application developers for everything from simple programs to advanced, enterprise-wide client/server applications thanks to its, different editions.

Professional Edition

The VB Professional Edition revolves around building Windows solutions quickly, making it a good fit for corporate IS departments. Coupled with the Visual Basic for Applications language engine (common across many MS apps), you get core developers able to communicate easily, leveraging their skills.

Enterprise Edition

For large programming teams building big applications, Microsoft provides Visual Basic, Enterprise Edition. It enables teams to develop with Visual Basic via the integrated Microsoft Visual SourceSafe(TM) project-oriented version control system. The Enterprise Edition also features fast, direct access to remote client/server databases with the Remote Data Control.

The list of what the Enterprise Edition of VB can do is long and impressive. Assuming you have at least half a brain, Microsoft expects that you can develop and debug a fully distributed application on a single workstation, then dynamically deploy it to network servers without much special help.

You can partition your application into MS-standard reusable OLE business rule components, which makes it easy to reuse these components, across your net, with different development tools. You can also use the Remote Data control for high-speed access to ODBC data sources such as Microsoft SQL Server and ORACLE. You can also keep team projects on track by using the Visual SourceSafe version control system which works through MS's Solutions Frameworks

Network for Distributed Services.

In addition, and in no small measure, VB was designed to work with future Windows OS architectures (including NT), making it a very forgiving environment when MS pushes all of its Win95 customers over to WinNT (Cairo) in 1997-98. The Enterprise Edition of VB touts itself as the first second generation client/server tool that encourages developers to build scalable, maintainable, and widely reusable applications in a RAD environment. While that may be an exaggeration, the Enterprise Edition of VB is an important tool that will be widely used by our Windows chums.

And we have nothing like it for the Macintosh.

Numbers Don't Lie

And that, like everything else in the VB scenario is bad for us. More than two million copies of VB have been put into developers hands, making it one of the most popular platform-specific tools you can buy for Windows. Name one Mac development tool that has sold two million copies.

The Microsoft Way

Bottom line, Microsoft is not about to expand VB's reach to help us. Instead, they're contented with keeping "the Mac in its place," as a senior MS official told me recently. He went on to say that there was "no chance at all," of VB being ported to the Mac, because MS was "not interested long term in helping the Mac work with Windows." In fact, since "we'd rather that you guys just go away, why should we port our best devtools to you," he concluded.

Why, indeed?

To receive information
on any products
advertised in this issue,
send your request
via Internet:
productinfo@xplain.com

By Jim Lloyd, jim@melongem.com

The OpenDoc Development Framework

A modern framework for OpenDoc development.

Developing for OpenDoc

The OpenDoc Development Framework (ODF) is a modern object-oriented framework for developing OpenDoc components for Macintosh and 32-bit Windows platforms. It significantly reduces the work required to create an OpenDoc component editor, especially if you're developing the part editor for both Macintosh and Windows.

In this article, I'll provide you with an overview of ODF. Because ODF is too large to address in one short article, I'm forced to condense and highlight. My goal is to provide you with an understanding of the problems ODF solves for you and how it solves them, and to give you a taste of what it is like to develop an OpenDoc component using ODF. I assume some level of understanding of OpenDoc, but not much, so if this is your first exposure to OpenDoc, I suggest you dive in anyway.

Why Another Framework?

Macintosh developers today have their choice of several frameworks. Some of the better known frameworks include Apple's MacApp, Symantec's TCL, Metrowerks' PowerPlant, MacTech's Sprocket, and Paul Dubois' TransSkel. If you've used any of these frameworks, then you probably appreciate the time and aggravation they can save you by providing a solid foundation on which to build your application. These frameworks not only provide you with code common to all Macintosh applications, they provide you with an architectural framework that makes it easier to design your own application. If all goes well, you simply "fill in the colors and connect the dots," so to speak.

The degree that "all goes well" depends on the framework you choose and your application's problem domain. The truth is, any framework is designed to work well with a specific range of application domains. If your application doesn't fit within that range, then the framework will at best get in your way from time to time, and at worst be totally useless. You are not likely to have much success trying to write a 2K INIT using TCL or MacApp. I don't consider this to be a shortcoming of those frameworks; common sense dictates that one shouldn't assume a given tool is the right tool for all jobs, big and small.

Of course, a framework shouldn't be a single tool; it should be a collection of tools that work well together. If the framework is done well, some of these tools will be applicable to any problem domain. Modern frameworks are designed with this in mind. PowerPlant, for example, is well suited for both simple "dashboard" applications and large-scale applications that use multiple document types, and pieces of the framework can be used in programs that don't fit these domains.

Why ODF?

Why ODF? Because even a flexible framework like PowerPlant isn't appropriate for all applications. First, developers need to

Jim Lloyd – Jim Lloyd has worked on an eclectic range of programming projects over the last 16 years, including image processing of radar images, collection and analysis of kinesiology data, income tax preparation, relational database management using associate array processors, and natural speech recognition. Since 1992 he has focused on cross-platform object-oriented frameworks. For the last year he has been a consultant to Apple Computer on the ODF project, and he recently accepted an offer to join Apple full time. Strange timing, since Jim is in the process of moving from Mountain View to a house he bought in San Francisco. Needless to say, Jim is already looking forward to the day that telecommuting at 128K or better becomes a reality.... You're welcome to contact him at jim@melongem.com, or jim_lloyd@powertalk.apple.com.

deliver their applications on multiple platforms, including Microsoft Windows. Second, OpenDoc introduces a new set of requirements that may be very difficult to fulfill unless the requirements are considered in the initial design of the framework. ODF addresses both of these issues. ODF is a modern framework designed to be flexible and robust, to be cross-platform, and to provide complete support for OpenDoc.

ODF is a cross-platform framework because its APIs abstract away the particulars of each operating systems' specific data structures and APIs. ODF defines platform-independent APIs, and implements these APIs on multiple operating systems. Developers who program to the framework's APIs exclusively should be able to move their applications to every operating system supported by the framework by simply recompiling their code. ODF currently supports the Macintosh OS for both 68K and PowerPC, and supports Microsoft Windows for its 32-bit operating systems, Windows 95 and Windows NT.

ODF includes a rich foundation of subsystems organized into two layers: the Foundation layer and the OS layer. Subsystems in the Foundation layer have no dependencies on the subsystems in the OS layer, and are generally useful for a very wide range of application domains. Subsystems in the OS layer provide operating system services such as files, resources, and graphics using platform-independent interfaces.

On top of these two layers we add a third layer, known as the Framework layer. The ODF Framework layer is specific to building OpenDoc component editors, and is designed to solve the problems of that particular domain as well as possible. ODF provides full coverage of the OpenDoc APIs, and correctly implements the human interface standards. If you use ODF to develop your OpenDoc part, you'll need to do less work to make your part functional. You'll also save yourself much of the additional effort required to make your part interoperate correctly with other parts, as well as the effort required to provide the correct human interface behavior.

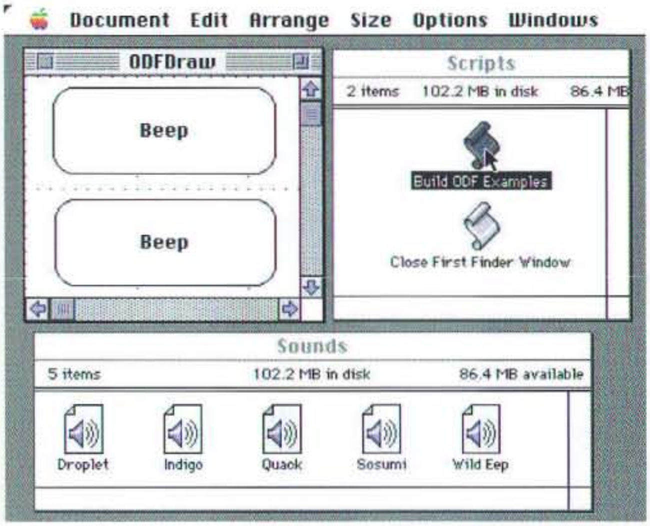
An extra benefit to using ODF is that it provides some of the best sample code for OpenDoc parts that you can find. The ODF examples range from the minimalist `ODFHello` part, which is our interpretation of "Hello World", implemented as a part, to `ODFDraw`, a cross-platform drawing part with floating palettes for drawing tools, pattern selection, and color selection. Of greater significance, `ODFDraw` is a robust container part for embedding other parts. `ODFDraw` has been the container of choice for OpenDoc demonstrations for much of the last year.

Also included among the ODF examples are a bitmap part, a clock part, a movie part, a table part, and finally the "beeper" part, which will play a big role in this article.

Extending The Beeper's Functionality

`ODFBeeper` is a simple button part that can play a sound resource. `ODFBeeper`, as delivered on previous releases of ODF, supports drag-and-drop so that you can change the sound that is played by dropping a sound file onto the button. In this article, we'll extend the Beeper part to also execute any compiled AppleScript dropped onto it. Thanks to Eric Jackson for coming up with this idea and drafting me to work on it with him at a recent ODF Coding Retreat. Eric and I did a quick hack so that the Beeper would execute scripts *instead* of playing a sound. It occurred to me later that with only a little more effort, the Beeper could do either sounds or scripts. The code presented in this article does both, and uses an architecture that would make it easy to add other kinds of actions besides sounds and scripts.

In order to see where we are going, let's first look at a screen snapshot of the final product:


Document Edit Arrange Size Options Windows
ODFDraw
Beep
Beep
Scripts
2 items 102.2 MB in disk 86.4 MB
Build ODF Examples
Close First Finder Window
Sounds
5 items 102.2 MB in disk 86.4 MB available
Droplet
Indigo
Quack
Sosumi
Wild Eep

ODFBeeper User Interface (such as there is)

This screen shot shows an `ODFDraw` document being used as a generic container. It contains two `ODFBeeper` parts embedded inside it. Conveniently located nearby are some sound files and some compiled AppleScript files. The user can drag any of these files onto either of the buttons. Clicking on the button will perform the action corresponding to the last file dropped onto the button. Changes to the whole document are persistent. The user can use the capabilities of the container application to rearrange the buttons, and can change the action of a button by dropping a new script or sound file onto the button. Saving the document saves all of the state, so the buttons will be in the same locations and perform the same actions the next time the document is opened.

Developing A Part Editor

Developing a part with ODF is much like developing an application with an application framework like PowerPlant, TCL, or MacApp. You can use your favorite C++ programming tools, including Apple's, Metrowerks' and Symantec's development environments, and other tools such as Object Master from ACIUS and The Debugger from Steve Jasik. The ODF view system is still evolving, so currently there are no visual tools for laying out your part's views, but visual tools for ODF are under development.

MACWORLD EXPO SAN FRANCISCO

The ultimate technology thrill awaits you as San Francisco comes alive at the world's largest and most comprehensive Macintosh-focused event of the year! MACWORLD Expo/San Francisco will bring you products so advanced they'll take your breath away. Experts so knowledgeable they'll easily solve your toughest questions. A glimpse at the future that's more reality than virtual. Kick off the new year by exploring the possibilities MACWORLD Expo has to offer.

MACWORLD Expo/San Francisco has got what you're looking for and what you need. Build on your power with informative conference sessions including networking, spreadsheet analysis, databases, education, research, word processing, accounting, law, CAD/CAM, forecasting, design, publishing, illustration, animation, music, programming, world wide web/internet, entertainment and multimedia.

Choose from dozens of conference sessions led by industry experts. Learn what's new and exciting at Apple . . . chart your own course through cyberspace . . .advance your knowledge of the hottest software . . . cash in on the latest innovations . . . share insider's tips from the world's savviest Macintosh users in the universe.

Mail in or fax the coupon below to get the latest brochure and find out how you can build on the power of MACWORLD Expo.

MACWORLD Expo/San Francisco
January 9-12, 1996 Moscone Center

get information on the WWW at:
http;//www.mha.com/macworldexpo

CALL 800-645-EXPO

Please send me more information on
MACWORLD Expo/San Francisco
I am interested in: □ Exhibiting □ Attending

Name ______________________

Title ______________________

Company ______________________

Address ______________________

City/State/Zip ______________________

Phone ______________ Fax ______________

Mail to: MHA Event Management, 1400 Providence Hwy., Norwood MA 02062
Or Fax to: 617-440-0357

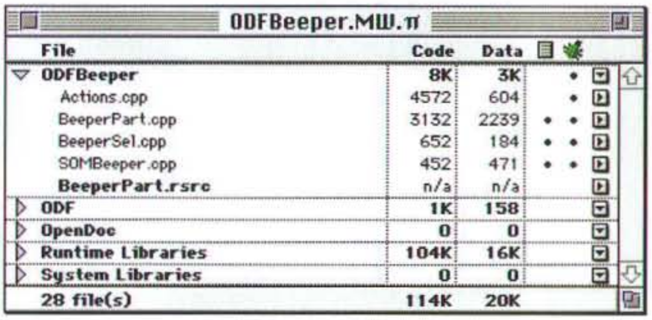
MacTech

For the purposes of this article, I started with the source for the ODFBeeper example part included with the 1.0d9 release of ODF, which was provided on the OpenDoc DR3 CD. If you have a source code disk subscription, the project and source for this extended example will be on your source code disk, but you'll still need the OpenDoc DR3 CD if you want to further enhance the example. If you don't have a source code subscription, keep an eye out for the OpenDoc DR4 release; ODF 1.0d11 will be on that CD and will include the extended ODFBeeper example.

By the way, there's no way that MacTech could publish all the source code necessary for this sample (and you wouldn't want to try to type it all in!), so I won't be showing complete source code listings in this article. Instead, I'll show the relevant snippets of code as they are discussed. In these snippets, I'll sometimes omit code not directly relevant to the discussion by replacing several lines of code with a single line containing an ellipsis and optionally a comment, like this:

```
... // irrelevant stuff omitted
```

ODF parts can be developed for either PowerPC or 68K using several different environments. For this article I used the PowerPC compiler in the CW6 release of Metrowerks CodeWarrior Gold and haven't yet tested the part with other environments, but you can expect the part to be buildable with all supported development environments on the DR4 CD. The following is a snapshot of the ODFBeeper CodeWarrior project:



ODFBeeper.MW.π

File	Code	Data		
▽ ODFBeeper	8K	3K		•
Actions.cpp	4572	604		•
BeeperPart.cpp	3132	2239	•	•
BeeperSel.cpp	652	184	•	•
SOMBeeper.cpp	452	471	•	•
BeeperPart.rsrc	n/a	n/a		
▷ ODF	1K	158		
▷ OpenDoc	0	0		
▷ Runtime Libraries	104K	16K		
▷ System Libraries	0	0		
28 file(s)	114K	20K		

ODFBeeper Project

The ODFBeeper example uses four source files. BeeperPart.cpp implements the classes CBeeperPart and CBeeperFrame, which are subclasses of the ODF classes FW_CPart and FW_CFrame respectively. BeeperSel.cpp implements the CBeeperSelection class, which is a subclass of the ODF class FW_CSelection. Actions.cpp implements the classes CAction, CSoundAction, and CScriptAction, all of which are completely specific to ODFBeeper; ODF itself knows nothing about actions. SOMBeeper.cpp contains the C++ bindings for the required SOM subclass of ODPart, which won't be discussed in this article. In addition, ODFBeeper uses the resource file BeeperPart.rsrc that contains a typical assortment of icons, strings, etc. It won't be discussed further in this article either.

I'll now proceed to describe the source code in the three source files Actions.cpp, BeeperPart.cpp, and BeeperSel.cpp, in that order. However, before discussing the Action classes, I think it would be worthwhile to partially discuss the CBeeperPart class. If you are familiar with the Model-View-Controller architecture from Smalltalk that has been adapted in various incarnations in most frameworks, then you can consider the Action classes to be the Model, CBeeperFrame to be the View, and CBeeperPart to be the Controller. Since the CBeeperPart is in control, it is worthwhile starting with its relationship to the Action classes.

CBeeperPart

Here is a partial definition of the class CBeeperPart:

```
class CAction; // forward declaration

class FW_CLASS_ATTR CBeeperPart : public FW_CPart
{
public:
   ... // constructors, methods inherited from FW_CPart

//——————————————
// New API
//
public:
   CAction*        GetAction() { return fAction; }
   void               SetAction(CAction* action);

public:
   void               DoAction();

//——————————————
// Data Members
//
private:
   CAction*        fAction;
};
```

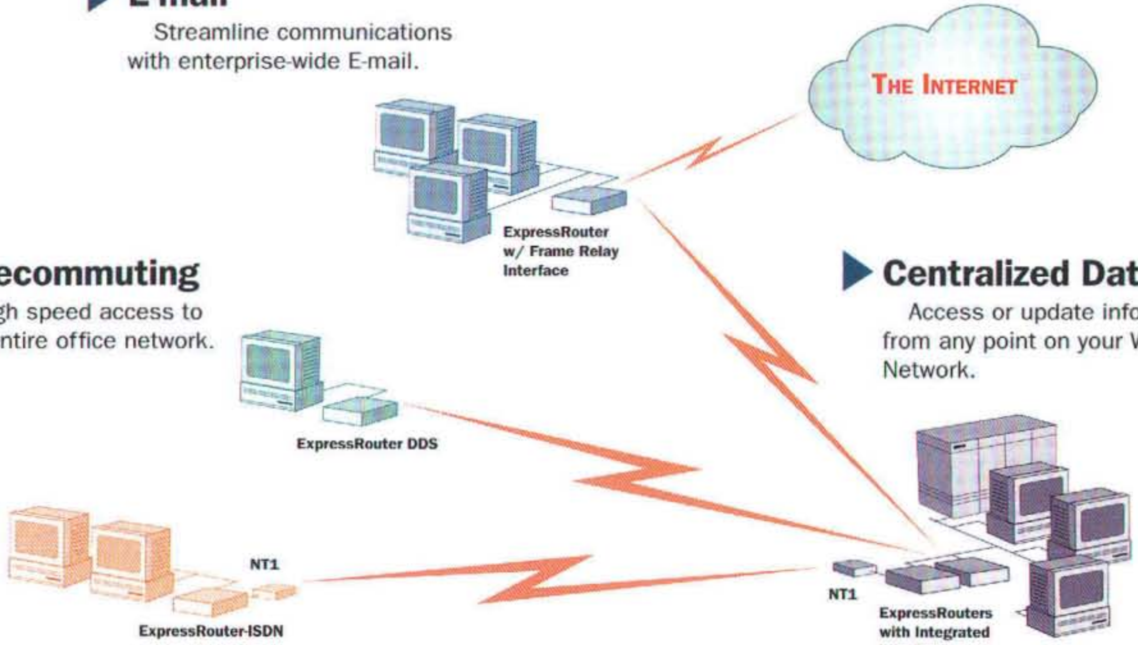
In ODF 1.0d9, CBeeperPart had a single data member that was a handle to the sound resource to play when the button was clicked. Since we want our new part to perform different kinds of actions, we've changed the data member to be a pointer to an abstract class CAction. We'll use the Action abstraction to minimize the dependencies of CBeeperPart on the details of implementing sound and script actions. CBeeperPart knows about the existence of abstract actions, and has a very simple protocol for actions which include the ability to get and set the current action, and to forward a request to apply the action to the current CAction object.

CAction

Let's now look at the definition of the abstract class CAction:

```
class FW_CLASS_ATTR CAction
{
//——————————————
// Initialization/Destruction
//
public:
   CAction();
   virtual ~CAction();

//——————————————
// Action protocol
```

Fast remote internetworking in a *single* box.


THE INTERNET
ExpressRouter w/ Frame Relay Interface
ExpressRouter DDS
NT1
ExpressRouter-ISDN
NT1
ExpressRouters with Integrated Interfaces

Shared Access to the Internet

Your whole department or organization can connect to the Internet—full time or just when you need to.

E-mail

Streamline communications with enterprise-wide E-mail.

Telecommuting

High speed access to the entire office network.

Centralized Databases

Access or update information from any point on your Wide Area Network.

Videoconferencing

Close the gap between distant sites with LAN-to-LAN videoconferencing.

Fast File Transfer

Transfer files at speeds up to 1.544Mbps—10 MB per minute.

What makes the ExpressRouter the best choice for your Wide Area Network (WAN) needs?

- Built-in WAN interfaces for Frame Relay , ISDN, T1, and 56Kbps DDS.
- Expandable architecture that supports up to 3 high-speed WAN ports.
- Built-in support for AppleTalk, TCP/IP and Novell IPX LAN protocols.
- Interoperability via PPP.
- Support for SNMP and Telnet management standards.
- Powerful, easy-to-use auto-dialer software designed for the Macintosh.

Engage Communication, Inc.
9053 Soquel Drive • Aptos, CA • 95003
Tel: 408.688.1021 • Fax: 408.688.1421 • E-mail: sales@engage.com •WWW: www.engage.com

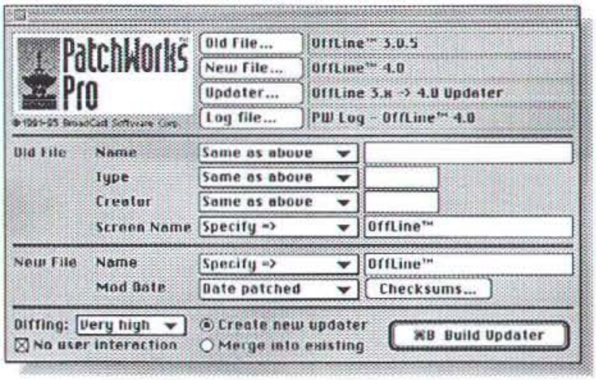
BUILD UPDATERS WITHOUT PROGRAMMING!

PatchWorks™ Pro

PatchWorks Pro has many options, but only one function: to create small, stand-alone Updater Applications which field-upgrade your products to newer versions. Updaters can be freely and safely distributed online, since they are only of use to end-users who've already purchased your software.

UPDATERS ARE FAST, SAFE & EASY FOR END-USERS

Your end-user just downloads and runs the Updater. The Updater prompts the user for the original software via a "Get File" dialog box (if not in the same folder as the Updater). When the user clicks the "Update" button, the original software is updated immediately.



Old File: OffLine™ 3.0.5
New File: OffLine™ 4.0
Updater: OffLine 3.x -> 4.0 Updater
Log file: PW Log - OffLine™ 4.0

Old File
Name: Same as above
Type: Same as above
Creator: Same as above
Screen Name: Specify => OffLine™

New File
Name: Specify => OffLine™
Mod Date: Date patched

Diffing: Very high
☒ No user interaction
● Create new updater
○ Merge into existing

UPDATERS ARE CREATED IN MINUTES!

Why tie-up valuable programming time building updater applications? Just fill in a dialog, and PatchWorks does the rest! Since there's no coding or scripting, no bugs are introduced.

- Works with apps, INITs, cdevs, fonts, drivers, etc.
- Works on resource and/or data forks
- Full support for PowerPC and fat binaries
- Updaters support multiple "old versions"
- Preserves personalization data (name, serial #, etc.)
- Customizable end-user interface
- Updaters can execute custom CODE resources
- Intelligent diff-ing produces small Updaters
- Distribution of Updaters is unrestricted and royalty-free!

NOW AN INDUSTRY STANDARD AFTER FOUR YEARS ON THE MARKET

PatchWorks Pro is used by Macintosh publishers such as: Apple Computer, Adobe Systems, MacroMind, Now Software, Symantec, WordPerfect, and many more. Pricing begins at $195. Download a fully-functional trial version from our web site: "http://www.broadcastsoft.com", or call us at (407) 241-0308.

BroadCast

BroadCast Software Corp
East: 407-241-0308, Fax: 407-241-3195
West: 503-317-0429, Fax: 503-317-0430
Net: http://www.broadcastsoft.com
Email: info@broadcastsoft.com

```
//
public:
  virtual void Internalize(Environment* ev,
              ODStorageUnit* storage) = 0;
  virtual void Externalize(Environment* ev,
              ODStorageUnit* storage) = 0;
  virtual void DoIt() = 0;
};
```

Action objects assume a two-step initialization; they are created in a default state and then initialized from a storage unit with the `Internalize` method. They also know how to save themselves to a storage unit via the `Externalize` method, and to do their action via the `DoIt` method. In this abstract base class, the three methods other than the constructor and destructor are pure virtual methods. Subclasses of `CAction` must provide a default constructor, a destructor, and implementations of the three virtual methods `Internalize`, `Externalize`, and `DoIt`.

CSoundAction

Let's now look at the implementation of the two Action subclasses. First, the definition for the class `CSoundAction`:

```
class FW_CLASS_ATTR CSoundAction : public CAction
{
//----------------------------------------
// Initialization/Destruction
//
public:
  CSoundAction();
  virtual ~CSoundAction();

  static FW_Boolean IsInStorage(
              Environment* ev, ODStorageUnit* storage);

//----------------------------------------
// Action protocol
//
public:
  virtual void Internalize(Environment* ev,
              ODStorageUnit* storage);
  virtual void Externalize(Environment* ev,
              ODStorageUnit* storage);
  virtual void DoIt();

//----------------------------------------
// Private implementation
//
private:
  void InternalizeSound(Environment* ev,
              ODStorageUnit* storage);
  void InternalizeSoundFile(Environment* ev,
              ODStorageUnit* storage);

  Handle fSoundHandle;
};
```

This class is a simple implementation of the class `CAction`, so the constructor, destructor, and three methods `Internalize`, `Externalize`, and `DoIt` are dictated by the base class. `CSoundAction` adds a public static method `IsInStorage`, two implementation methods for internalizing from specific sound formats, and a data member for storing a sound handle. Let's look at the definitions of the methods of this class, beginning wit the implementation of the `Internalize` method inherited from `CAction`:

CSoundAction::Internalize

```
void CSoundAction::Internalize(Environment* ev,
                ODStorageUnit* storage)
{
  FW_Boolean internalized = FALSE;

  if (storage->Exists(ev,
          kODPropContents, kSoundScrapKind, 0))
  {
    // Mac 'snd ' in Scrap
    storage->Focus(ev,
              kODPropContents,
              kODPosUndefined,
              kSoundScrapKind,
              0,
              kODPosUndefined);
    InternalizeSound(ev, storage);
  }
  else if (storage->Exists(ev,
          kODPropContents, kSoundFileKind, 0))
  {
    // Mac sound file
    storage->Focus(ev,
              kODPropContents,
              kODPosUndefined,
              kHFSFlavorType,
              0,
              kODPosUndefined);
    InternalizeSoundFile(ev, storage);
  }
}
```

This code examines the storage unit for the existence of either a sound resource (OSType `'snd '`) or a sound file (OSType `'sfil'`). If either is found, the code focuses the storage unit and then calls one of the two private member functions for reading the specific data type.

By the way, the constants `kSoundScrapKind`, `kSoundFileKind`, and `kHFSFlavorType` are strings defined in the header file BeeperDef.h using a naming convention agreed upon by the OpenDoc implementors (Apple, IBM, and Novell):

```
#define kHFSFlavorType "Apple:OSType:Scrap:hfs "
#define kSoundScrapKind "Apple:OSType:Scrap:snd "
#define kSoundFileKind  "Apple:OSType:FileType:sfil"
```

If `Internalize` detects a sound resource in the scrap, it calls the method `InternalizeSound`:

CSoundAction::InternalizeSound

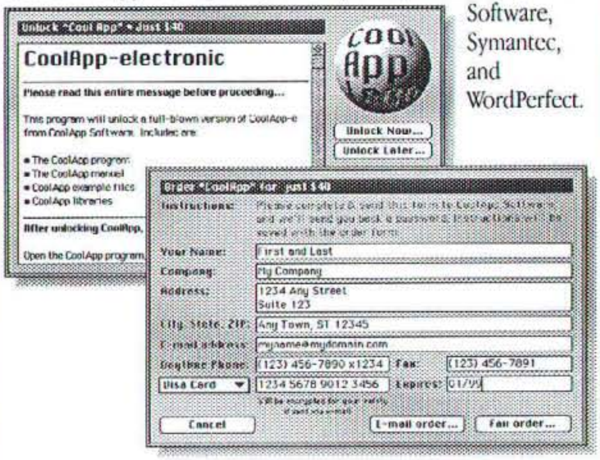
```
void CSoundAction::InternalizeSound(Environment* ev,
                ODStorageUnit* storage)
{
  // Assume the storage unit is already focused
  unsigned long size = storage->GetSize(ev);
  if (size > 0)
  {
    FW_CAcquireTemporarySystemHandle handle(size);
    FW_CByteArray byteArray;
    storage->GetValue(ev, size, byteArray);
    byteArray.CopyBuffer(handle.GetPointer(), size);
    fSoundHandle = handle.GetPlatformHandle();
    handle.Orphan();
  }
}
```

`InternalizeSound` assumes the entire content of the property value is a sound resource. It allocates a handle of the appropriate size and then reads the data into the handle. Finally, it stores the handle in the `fSoundHandle` field. Looking at the code, we see that a little more is going on here, which is worth explaining in some detail.

SOLID SOLUTION FOR
ELECTRONIC DISTRIBUTION

BroadCast™

BroadCast is a highly secure, patent-pending system for safely distributing "locked" software online, or on CD-ROM. This cutting-edge technology was developed by the same engineering team behind PatchWorks— our tool for creating online updaters that's become an industry standard used by clients like Apple Computer, Adobe Systems, MacroMind, Now Software, Symantec, and WordPerfect.



Unlock "Cool App" • Just $40

CoolApp-electronic

Please read this entire message before proceeding...

This program will unlock a full-blown version of CoolApp-e from CoolApp Software. Includes are:

- The CoolApp program
- The CoolApp manual
- CoolApp example files
- CoolApp libraries

After unlocking CoolApp,

Open the CoolApp program,

Order "CoolApp" for just $40

Instructions: [illegible]

Your Name: First and Last
Company: My Company
Address: 1234 Any Street
Suite 123
City, State, ZIP: Any Town, ST 12345
E-mail address: myname@mydomain.com
Daytime Phone: (123) 456-7890 x1234 Fax: (123) 456-7891
Visa Card 1234 5678 9012 3456 Expires: 01/99

BROADCAST IS EASY FOR YOU AND YOUR CUSTOMERS

In seconds, BroadCast securely locks your product into an Unlocker application that can be safely distributed to the public. To buy your product, customers simply run the Unlocker— then either call you, or complete an onscreen order form that's emailed or faxed to you with a unique "control number" (and encrypted credit card data).

Your staff processes the credit card, registers your new user, and provides your customer with a password based upon the control number. To unlock your software, the customer enters this password in the Unlocker.

BROADCAST IS INEXPENSIVE

We aren't resellers or distributors. Customers buy directly from you, and we take no percentage of your sales— just a small fee per transaction. In fact, BroadCast is distributed for free. We're betting you'll quickly use the free password points that come with BroadCast, and purchase more as your sales soar.

We'll even help attract customers by advertising your products for free on our "Software Unboxed" internet mall.

GET THE WHOLE STORY

We invite you to learn more by visiting our internet web site at "http://www.broadcastsoft.com", and download your free BroadCast Starter Kit, or email us: "info@broadcastsoft.com".

BroadCast

BroadCast Software Corp
East: 407-241-0308, Fax: 407-241-3195
West: 503-317-0429, Fax: 503-317-0430
Net: http://www.broadcastsoft.com
Email: info@broadcastsoft.com

Presenting the extra-strength text editor.

If you thought previous versions of QUED/M were powerful, wait until you program with QUED/M 2.7, loaded with these new pain-relieving features:

- Integrated support for THINK Project Manager™ 6.0 & 7.0
- THINK™ debugger support
- CodeWarrior™ support (now it's easier than ever to program for the Power Macintosh®!)
- MPW ToolServer™ support
- PopUpFuncs™ support
- Frontier™ Do Script support

For quick relief, call 800-309-0355 today.

Of course, QUED/M 2.7 still has all the features that make it easier to use than any other text editor:

- Macro Language lets you automate tedious tasks
- Search and Replace through multiple unopened files and using GREP metacharacters
- File comparisons using GNU Diff
- Unlimited undos and redos
- 10 Clipboards that can be edited, saved, and printed
- Customizable menu keys
- Text folding
- Display text as ASCII codes
- Plus many more features!

Get even more relief: Try QUED/M 2.7 now for just $69! You'll get a 30-day money back guarantee too! Call now to order. 800-309-0355.

NISUS Software Inc.

QUED/M is a trademark of Paragon Concepts, Inc. All other products are trademarks or registered trademarks of their respective holders.

107 S. Cedros Ave. • Solana Beach, CA 92075 • Tel (619) 481-1477 • Fax (619) 481-6154

To allocate the handle, the code uses an instance of the class FW_CAcquireTemporarySystemHandle, which is a resource acquisition helper object that allocates a "temporary" handle. The handle is considered to be temporary because unless instructed otherwise, the handle will be disposed when the helper object goes out of scope. On the Macintosh, the constructor of FW_CAcquireTemporarySystemHandle will allocate a handle in Multifinder temporary memory (throwing an exception if the allocation fails), and then lock the handle down in memory. On other platforms, it does something equivalent.

The FW_CAcquireTemporarySystemHandle destructor will unlock the object, and optionally dispose the handle. The default behavior is to dispose the handle, but calling the Orphan method instructs the helper object to not dispose the handle. This may seem a little involved, but it is a very convenient idiom for allocation of resources (like memory) that must be disposed if an exception is thrown before the resource is completely initialized.

In CSoundAction::InternalizeSound, we can assume that any of the three lines of code starting with the call to the method GetValue and ending with the call to the method GetPlatformHandle could throw an exception. In actual fact, with the current implementations of these functions, only the GetValue method could throw an exception, but in this case it hurts nothing to be conservative.

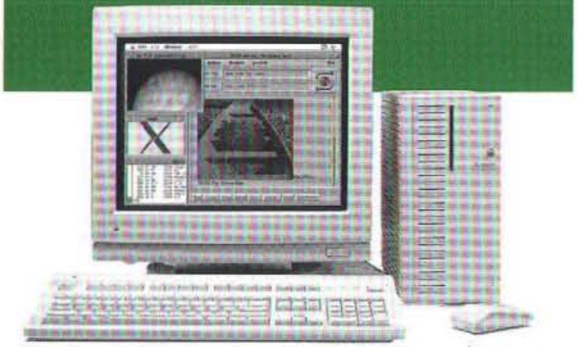
ODF provides a range of utility functions and helper objects that make it possible to write portable code that works with exceptions correctly. In the case of CSoundAction::InternalizeSound, the advantages of the platform-independence of the helper objects FW_CAcquireTemporarySystemHandle and FW_CByteArray aren't fully realized because the function as a whole is making an assumption specific to the Macintosh: 'snd ' resources exist only on the Macintosh. With a little extra effort, it would be possible to create an abstraction for sound data that would work on both Macintosh and Windows. Such a class will likely make it into ODF someday; in the meantime, we leave it as an exercise for the reader.

Let's now look at the method to internalize a sound file:

CSoundAction::InternalizeSoundFile

```
void CSoundAction::InternalizeSoundFile(Environment* ev,
          ODStorageUnit* storage)
{
  // Storage Unit should already be Focused on HFSFlavor
  unsigned long hfsSize = storage->GetSize(ev);
  if (hfsSize > 0)
  {
    // Get the HFS flavor
    HFSFlavor hfsInfo;
    FW_CByteArray byteArray;
    storage->GetValue(ev, sizeof(HFSFlavor), byteArray);
    byteArray.CopyBuffer(&hfsInfo, sizeof(HFSFlavor));

    // Create platform independent file spec
    FW_CFileSpecification fileSpec(hfsInfo.fileSpec);
```

Now your computer can be both a Macintosh and a UNIX workstation.



The Mach^Ten operating system combines the high-level functionality of a networked UNIX workstation with Macintosh's wide array of applications.

Any NFS server can be used to store Macintosh files. Users can access them by double clicking as on the local disk.

Mach^Ten is a Berkeley BSD UNIX that runs on the Classic to the Power Macintosh, including PowerBooks and Duos! So in addition to all of the easy-to-use applications that make Macintosh one of the most personable computers around, you get a MACH-based UNIX with pre-emptive multitasking.

Mach^Ten's strength lies in the way it extends the Macintosh Operating System with UNIX networking and software development tools. The Macintosh/UNIX integration is so strong that you can even use Mac programs & utilities on UNIX data, and UNIX programs & utilities on Mac files.

Full internet protocol support ensures fast, easy client and server NFS, electronic mail, and file transfer between the Mac and all TCP-based entities on your network.

The UNIX software development system includes the GNU C and C++ compilers and libraries to let you create new applications or port existing ones. The Motif toolkit and suite of X clients and X client libraries make developing distributed applications a breeze.

And Tenon's high performance X Server lets you use your Macintosh or Power Macintosh as an X terminal.

Join the many satisfied users of proven, reliable Mach^Ten UNIX, and start turning all of your Macs into open systems today!

For more information, or to order
call 1-800-6-MACH-10.

Internet: info@tenon.com
http: //www.tenon.com

TENON INTERSYSTEMS *New Dimensions in Personal Workstation Technology*

Tenon Intersystems 1123 Chapala Street Santa Barbara, CA 93101 Tel: 805-963-6983 Fax: 805-962-8202

©1994 Tenon Intersystems. The Tenon Intersystems name and Mach^Ten are trademarks of Tenon Intersystems. Macintosh, Classic, PowerBook, Power Macintosh and Duo are registered trademarks of Apple Computer, Inc. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

```
    // Open the resource file
    FW_CResourceFile resourceFile(fileSpec);

    // Get any (i.e. the first) sound resource
    const FW_ResourceType resourceType = 'snd ';
    const FW_ResourceId resourceId = 1;
    Handle sound = ::Get1IndResource(resourceType,
                        resourceId);

    if (sound != NULL)
    {
      // if succeeded, detach the handle and keep it
      ::DetachResource(sound);
      fSoundHandle = sound;
    }
    else
    {
      // No sound resource loaded, throw an exception
      FW_THROW(FW_XResourceNotFound(resourceFile,
                        resourceId,
                        resourceType));
    }
  }
}
```

InternalizeSoundFile is similar to InternalizeSound. The difference is that instead of getting the sound resource directly from the storage unit, it instead gets an HFS file specification from the storage unit. It then uses this file specification to open the file and grab the sound resource from the file. The code here is a bit complicated because sound files don't use fixed resource IDs, so the code must look for *any* sound resource. Unfortunately, this is a function that we don't (yet!) provide in the Resources subsystem of ODF, so we have to resort to direct calls to the Macintosh toolbox resource manager. The good news is that ODF doesn't get in the way of doing this. In fact, it is still convenient to use an instance of the class FW_CResourceFile to open the file, since it will assume responsibility for closing the file when it goes out of scope. So, we use ODF to create and open the resource file, and then use the two toolbox calls Get1IndResource and DetachResource to get the sound resource from the file. Since we've taken direct responsibility for reading the resource, it is our responsibility to handle any errors, which we do by throwing the appropriate exception.

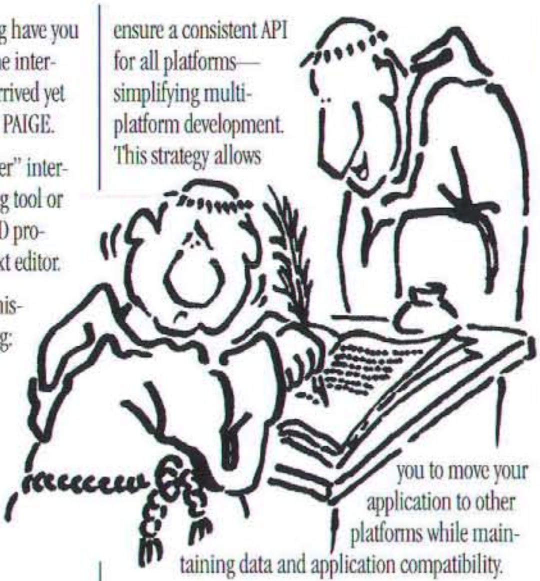
By the way, a new feature already added since ODF 1.0d9 will make it simpler to throw standard exceptions. Instead of writing the above FW_THROW statement, you could instead write something like this:

```
FW_FailOnError(::ResError());
```

We've seen how to internalize sound resources, now let's see the code to externalize them:

CSoundAction::Externalize

```
void CSoundAction::Externalize(Environment* ev,
```



When it comes to text editing have you ever hoped for a little divine intervention? Well if it hasn't arrived yet you should be considering PAIGE.

Whether you're developing the next "killer" internet application, building a custom publishing tool or adding advanced word processing to your CAD program, you can't ignore the quality of your text editor.

With PAIGE you can create the most sophisticated text features in the business. Including:

- Stylized Text
- Text Wrapping
- Embedded Objects
- Shapes & Containers
- Virtual Memory
- Style Sheet Support
- Multi-lingual Capable
- Portable C Code
- Royalty Free

PAIGE is written in portable C and uses no global variables. Machine specific code is minimized to ensure a consistent API for all platforms—simplifying multi-platform development. This strategy allows you to move your application to other platforms while maintaining data and application compatibility.

To simplify the integration of PAIGE into Windows (16-bit, 32-bit, '95 or NT) applications, our library ships as DLL's and can be used as a custom control. Combined with the message based demo source code and simple introduction guide, PAIGE gets programmers up and running quickly.

So why should you buy PAIGE? TIME. Programmers can't afford to reinvent the wheel when implementing advanced text features within their applications.

Join the hundreds of major software publishers using PAIGE as their total text solution. For a complete technical and pricing summary, or to request our product demo, contact the DSI sales department today at **800-327-6703** or **360-573-9155**.

DataPak Software Inc.

9317 NE Hwy 99, #G • Vancouver, WA 98665-8900
Bus: (360) 573-9155 • Fax: (360) 573-9269

Internet: 76424.3027@compuserve.com
www.teleport.com/~datapak/datapak.html
AppleLink: D0142 • AOL: DATAPAK1
• CS: 76424,3027

"Say, nice job brother John. Now could you make it a three column format with digital memos embedded in the text stream?"

```
                ODStorageUnit* storage)
{
  // Assume we're focused on the correct property
  storage->AddValue(ev, kSoundScrapKind);
  if (fSoundHandle != NULL)
  {
    FW_CAcquireLockedSystemHandle lock(fSoundHandle);
    FW_CByteArray byteArray(*fSoundHandle,
          FW_CMemoryManager::
          GetSystemHandleSize(fSoundHandle));
    storage->SetValue(ev, byteArray);
  }
}
```

Externalize simply adds a sound scrap property value, and then writes out the sound handle. Since the handle needs to be locked in order to be written out, we take advantage of the FW_CAcquireLockedSystemHandle helper object to lock the object and assume responsibility for unlocking it. The rest of the code should be self-explanatory.

Finally, we round out our implementation of CSoundAction with the DoIt method:

CSoundAction::DoIt

```
void CSoundAction::DoIt()
{
  if (fSoundHandle != NULL)
  {
    FW_CAcquireLockedSystemHandle lock(fSoundHandle);
    ::SndPlay(NULL, (SndListHandle)fSoundHandle, TRUE);
  }
}
```

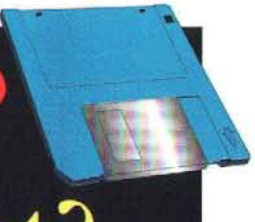
Here we simply lock the handle with a helper object and then use the toolbox function SndPlay to play it. What could be easier?

CScriptAction

The implementation of the class CScriptAction has a lot in common with the class CSoundAction. Compiled scripts, like sounds, are stored in memory as relocatable blocks accessed via handles, and are stored on disk as resources in the resource fork. The code to internalize and externalize a compiled script is therefore similar to the code to internalize and externalize a sound resource. The only significant difference is in the implementation of InternalizeScriptFile, so for the sake of brevity, we'll skip the code for CScriptAction::InternalizeScript and CScriptAction::Externalize.

CScriptAction::InternalizeScriptFile

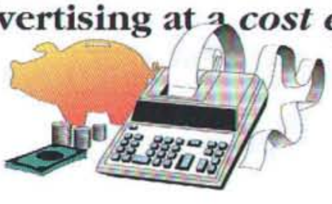
```
void CScriptAction::InternalizeScriptFile(Environment* ev,
                ODStorageUnit* storage)
{
  // Storage Unit should already be Focused on HFSFlavor
  unsigned long hfsSize = storage->GetSize(ev);
  if (hfsSize > 0)
  {
    // Get the HFS flavor
    HFSFlavor hfsInfo;
    FW_CByteArray byteArray;
    storage->GetValue(ev, sizeof(HFSFlavor), byteArray);
```



Got Software for Sale?
No Advertising Budget?

Try listing your product in MacTech Magazine's Mail Order Store

Classified Advertising at a *cost effective* rate!



For more information, call, fax or e-mail:
Voice: 805/494-9797 • Fax: 805/494-9798
Internet: marketing@xplain.com

For Macintosh Programmers & Developers
MacTech™ MAGAZINE

```
    byteArray.CopyBuffer(&hfsInfo, sizeof(HFSFlavor));

    // Make the file specification
    FW_CFileSpecification fileSpec(hfsInfo.fileSpec);

    // Acquire an opened resource file,
    // will be closed automatically
    FW_CResourceFile resourceFile(fileSpec);

    // Acquire an opened resource,
    // will be released automatically
    FW_CResource resource(resourceFile, 128, 'scpt');

    // Copy the resource handle
    fScriptHandle = FW_CMemoryManager::
          CopySystemHandle(resource.GetHandle());
  }
}
```

Compiled scripts are always stored with the fixed ID 128. This allows us to use the ODF resource subsystem to load the resource instead of reverting to the MacOS toolbox. This is done by simply creating a instance of the `FW_CResource` class. This class loads the resource in its constructor, and releases the resource in its destructor. It is exception-safe, and works correctly on both Macintosh and Windows. All we need to do to finish internalizing the script is to copy the resource handle and store it inside the `fScriptHandle` data member.

The `CSoundAction::DoIt` method executes the script using the static method `CScriptAction::LoadAndExecuteScript`:

CScriptAction::DoIt

```
void CScriptAction::DoIt()
{
  if (fScriptHandle != NULL)
    LoadAndExecuteScript(fScriptHandle);
}
```

CScriptAction::LoadAndExecuteScript

```
void CScriptAction::LoadAndExecuteScript(Handle scriptData)
{
  AEDesc scriptDesc;
  OSAID scriptID, resultID;
  AEDesc scriptText;
  OSAError error;
  static ComponentInstance gComponent = 0;

  if (!gComponent)
    gComponent = ::OpenDefaultComponent(
          kOSAComponentType,
          kOSAGenericScriptingComponentSubtype);

  // load the script data
  scriptDesc.descriptorType = typeOSAGenericStorage;
  scriptDesc.dataHandle = scriptData;
  error = ::OSALoad(gComponent, &scriptDesc,
              kOSAModeNull, &scriptID);

  if (error == noErr)
  {
    // execute the compiled script in the default context
    error = ::OSAExecute(gComponent,
            scriptID, kOSANullScript,
            kOSAModeNull, &resultID);
    error = ::OSADispose(gComponent, scriptID);
    error = ::OSADispose(gComponent, resultID);
  }
}
```

There's not much to say about this code. `LoadAndExceцuteScript` is lifted almost verbatim from *IM:IAC*, pp. 10–16. I haven't bothered to address the fact that it just lets errors fall on the floor unnoticed, though with the post-1.0d9 additions to exception handling it will be a relatively easy task. All that is necessary is to sprinkle a few calls to `FW_FailOnError(error)`.

That basically completes the description of the action classes. Observant readers will note that I've not yet described the two static methods `IsInStorage`, one each for the classes `CSoundAction` and `CScriptAction`. We'll get to that shortly. Let's move on to the implementation of the class `CBeeperPart`.

CBeeperPart

Earlier, we saw a partial class definition for `CBeeperPart`. At that time, we were interested in the protocol inherited from `FW_CPart`; we were only concerned with the protocol related to Actions. Now that we're ready to talk about the part protocol, it might be worthwhile to see a complete definition for the class `CBeeperPart`:

```
class FW_CLASS_ATTR CBeeperPart : public FW_CPart
{
public:
  static const ODValueType kPartKind;

//--------------------
// Initialization/Destruction
//
public:
  CBeeperPart(ODPart* odPart);
  virtual ~CBeeperPart();

  virtual void Initialize(Environment* ev);

//--------------------
// Inherited API
//
public:
  virtual FW_CFrame* NewFrame(Environment* ev,
              ODFrame* odFrame,
              FW_CPresentation* presentation,
              FW_Boolean fromStorage);

  virtual void InternalizeContent(Environment* ev,
              ODStorageUnit* storage,
              FW_CCloneInfo* cloneInfo);

  virtual void ExternalizeContent(Environment* ev,
              ODStorageUnit* storage,
              FW_CCloneInfo* cloneInfo);

//--------------------
// New API
//
public:
    CAction*      GetAction() { return fAction; }
    void            SetAction(CAction* action);

public:
    void            DoAction();

//--------------------
// Data Members
//
private:
  CAction*      fAction;
};
```

Let's begin with the constructor and destructor of CBeeperPart:

CBeeperPart::CBeeperPart

```
CBeeperPart::CBeeperPart(ODPart* odPart) :
  FW_CPart(odPart, CBeeperPart::kPartKind),
  fAction(NULL)
{
}
```

CBeeperPart::~CBeeperPart

```
CBeeperPart::~CBeeperPart()
{
  delete fAction;
}
```

The constructor initializes the `FW_CPart` base class, and initializes the `fAction` field to `NULL`. The destructor simply deletes the object pointed to by `fAction`. Of course, the `FW_CPart` constructor does quite a bit of work; `CBeeperPart` inherits a lot of default behavior from its `FW_CPart` base class. It is beyond the scope of this article to go into the details of the class `FW_CPart`, but suffice it to say that ODF makes it pretty easy to create a new part by overriding just a few methods, as we're about to see. The next method is the `Initialize` method:

CBeeperPart::Initialize

```
void CBeeperPart::Initialize(Environment* ev)
{
  FW_CPart::Initialize(ev);

  // — Register our Presentation
  CBeeperSelection *selection =
            new CBeeperSelection(ev, this);
  RegisterPresentation(ev, kODFBeeperPresentation,
            TRUE, selection);
}
```

To initialize the `CBeeperPart`, we must first call the inherited method to initialize the base class. Next, we create an instance of `CBeeperSelection`, which is a subclass of `FW_CSelection`. Selection objects are used to designate the user's selection for data interchange via any of the interchange mechanisms: copy/paste, drag/drop, or linking. Finally, we register a presentation. Parts may have multiple presentations for displaying their data content. A classic example is a spreadsheet application that can display its data in a tabular array, bar chart, pie chart, etc. Each presentation may have its own selection, so the selection object is owned by the presentation.

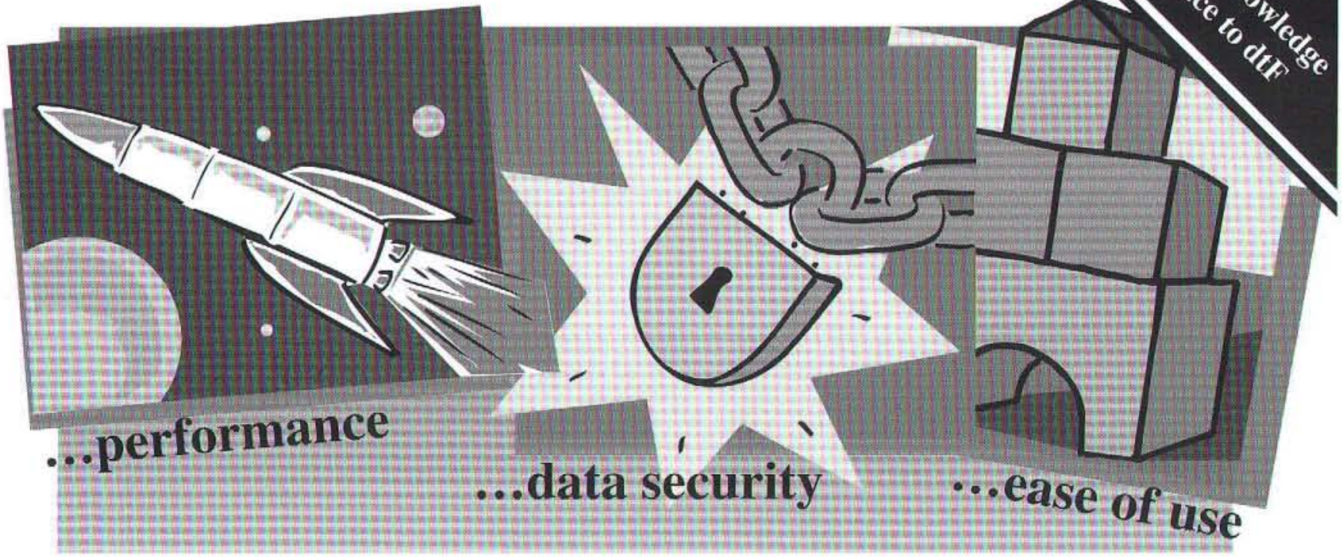
CBeeperPart::NewFrame

```
FW_CFrame* CBeeperPart::NewFrame(Environment* ev,
            ODFrame* frame,
            FW_CPresentation* presentation,
            FW_Boolean storage)
{
  return new CBeeperFrame(ev, frame, presentation, this);
}
```

`NewFrame` is a factory method which will be called whenever the part needs a new display frame. If a part supports multiple presentations it would create different kinds of frames, but our Beeper part supports only one simple presentation and creates only one kind of frame.

dtF The Relational Database System

New!
DataScript™ by General Knowledge
The AppleScript-Interface to dtF



When it comes to performance **dtF** is in a class all its own. **dtF** utilizes a proprietary query optimization and caching scheme to obtain unparalleled performance. And if you are in need of a quantum leap in performance, use the **dtF** native PowerPC standalone or server version.

Full transaction control and error recovery guarantee maximum data protection even after sudden system crashes. **dtF** databases are compressed and encrypted to protect against all unauthorized access, even disk editors.

Integrated data dictionary, security, automatic index selection, query optimization, deadlock detection and error recovery allow you to concentrate on your application. The **dtF** API was designed for a seamless integration with third-party toolkits and class libraries.

dtF features include SQL, BLOB support, electronic documentation, sample applications, tutorials, and tools for importing, exporting, creating, managing databases and users. **dtF** is currently available for Macintosh, DOS, Windows, OS/2. The C/C++ API is identical across all platforms.

Available for Macintosh System 7.x and native PowerPC. MPW C/C++ (68K and native PPC), Symantec C/C++, Metrowerks CodeWarrior (68K and native PPC). Separate versions for use with HyperCard, SuperCard, SmalltalkAgents® and Prograph CPX available.

Standalone applications built with dtF are royalty-free.

dtF Americas, Inc.
19672 Stevens Creek Blvd.,
Suite 128
Cupertino, CA 95014, USA

Phone: (800) DTF-1790
Fax: (510) 828-8755
AppleLink: DTF.AMERICA

dtF The Relational Database System

Share your vision—and your workload with an understanding, creative partner who can expedite all or part of the development process for you. The SU5 Group has a track record of creating quality products, ready to publish and market.

Making Your Ideas Come Alive

SU5

PHONE: 817 / 870-2557
INTERNET: sufive@onramp.net
WORLD WIDE WEB: http://www.rahul.net/sufive/

MACINTOSH SOFTWARE DEVELOPMENT

CBeeperPart::InternalizeContent

```
void CBeeperPart::InternalizeContent(Environment *ev,
                ODStorageUnit* storage,
                FW_CCloneInfo* cloneInfo)
{
  FW_ASSERT(fAction == NULL);

  if (CScriptAction::IsInStorage(ev, storage))
    fAction = new CScriptAction();
  else if (CSoundAction::IsInStorage(ev, storage))
    fAction = new CSoundAction();

  if (fAction != NULL)
    fAction->Internalize(ev, storage);
}
```

InternalizeContent is called when the part is being initialized from storage. CBeeperPart determines whether the storage contains a script or a sound, creates the appropriate action object, and then delegates to the action object. Here we finally see the purpose of the two static methods CScriptAction::IsInStorage and CSoundAction::IsInStorage. These methods examine the storage unit to see if they contain the appropriate data type.

This mechanism for creating an instance of the right action type based upon the contents of the storage unit is a bit simplistic. If we had many action kinds, it would become awkward, and a more extensible mechanism would become desirable. One reasonable choice would be based upon a dictionary mapping known data interchange types to factory functions. However, for the purpose of this example, such an approach is clearly overkill.

Once an action instance is created, we initialize it from the storage unit using the Internalize virtual function.

CBeeperPart::ExternalizeContent

```
void CBeeperPart::ExternalizeContent(Environment *ev,
                ODStorageUnit* storage,
                FW_CCloneInfo* cloneInfo)
{
  // If any kind of property content exists...
  if (storage->Exists(ev, kODPropContents, kODNULL, 0))
  {
    storage->Focus(ev, kODPropContents,
          kODPosUndefined, kODNULL, 0, kODPosAll);
    storage->Remove(ev);
  }
  storage->AddProperty(ev, kODPropContents);
  if (fAction != NULL)
    fAction->Externalize(ev, storage);
}
```

ExternalizeContent checks to see if any previous content exists in the storage unit, and if so, removes the old data. It then adds a content property, and delegates the task of externalizing to the Action object. This will result in a call to one of the two methods CSoundAction::Externalize or CSoundAction::Internalize, which we have already seen.

CBeeperPart::DoAction

```
void CBeeperPart::DoAction()
{
  if (fAction != NULL)
    fAction->DoIt();
}
```

DoAction is executed whenever the button is pressed via a mechanism that we'll discuss shortly. DoAction merely forwards the request to the current action object.

CBeeperPart::SetAction

```
void CBeeperPart::SetAction(CAction *action)
{
  if (action && action!=fAction)
  {
    delete fAction;
    fAction = action;
  }
}
```

SetAction is used to change the action object. It is called when a client of CBeeperPart wants to change the part's current action. We'll see it used by the CBeeperSelection class.

CBeeperFrame

ODF contains a class FW_CFrame which provides a rich definition of the behavior of OpenDoc frames. FW_CFrame inherits behavior from four mixin classes (FW_MEventHandler, FW_MGadgetContainer, FW_MDragDroppable, and FW_MIdle), and has about 100 methods! Despite the size and apparent complexity of FW_CFrame, implementing a subclass is relatively easy; CBeeperFrame overrides only five methods:

If you have a CD-ROM drive, then *you've gotta have* every article published in the first 10 years of MacTech Magazine!!

Available Now!!

... now in THINK Reference format!

MacTech CD-ROM, Volumes 1-10:

Over 1230 Articles.
All 115 Issues, including all of 1994.
All the Source Code.
THINK Reference 2.0.
Working Applications.
Full Documentation.
Demos for Developers.
And More!!

"When I designed THINK Reference, I envisioned endless databases at my fingertips. MacTech has doubled the information that is just a mouse click away."

– *Darrell LeBlanc, Formerly of Symantec,*
Author, THINK Reference 2.0

"The CD strikes me as an impressive and very useful resource. The only disadvantage I've found is that each answer the databases yield exposes me to so many more issues, that I find myself exploring the articles for the sheer wonder of it, and thus putting off the real coding I should be doing. :-)"

– *Nicholas De Mello*
MacTech CD Beta Tester

For Macintosh Programmers & Developers

MacTech™ MAGAZINE

Voice: 805/494-9797 • Fax: 805/494-9798
Internet: custservice@xplain.com
eWorld: MT.CustSvc • AppleLink: MT.CUSTSVC
CompuServe: 71333,1063
America Online & GEnie: MACTECHMAG

✓ **50 MB of MacTech Magazine articles, Volumes 1-10**

Every article, 1230+ of them, from all 115 issues of MacTech Magazine printed from 1984 through 1994. Articles ranging from Assembly to BASIC, C to Pascal, Forth to FORTRAN and more. *And the articles are in THINK Reference!*

✓ **Hyperlinks to *Inside Macintosh* Databases of THINK Reference**

The articles have hyperlinks to relevant portions of the *Inside Macintosh* databases of THINK Reference. For example, if you are looking for information on Aliases, look at the MacTech articles on Aliases and use the hyperlinks to jump to the *Inside Macintosh* entry for the Alias Manager. And now, with the articles in THINK Reference, you can do free-text searches 8-10 times faster than you could previously with On Location™ 2.0.

✓ **100+ MB of MacTech Magazine source code examples, samples, and utilities.**

These are the files that go with the magazine – the code that the articles are talking about. Use them in your own applications, with no royalties!

✓ **A fully-capable version of THINK Reference**

Including *Inside Macintosh* Volumes 1-6 (7 MB).

✓ **Sprocket – MacTech's Tiny Framework.**

Build your simple application using a lean, mean application framework. Experiment with new code without having to build a whole application around it!

✓ **80 MB of FrameWorks, MacApp®, MADA and SFA articles, files and source code.**

The most complete set of FrameWorks archives known.

✓ **Apple APIs, Utilities, and SDKs**

Including: Universal Header Files, Discipline, Macintosh Drag and Drop SDK, MacsBug, Telephone Manager SDK, Thread Manager SDK, TrueEdit 1.8 and and more.

FREE!
THINK™ Reference

Symantec's THINK Reference 2.0. Complete on-line guide to Inside Macintosh, Vol. I-VI, with cross referenced index, detailed information of each function, procedure and detail needed when programming the Macintosh.

SYMANTEC.™

✓ **Ariel Publishing's *Inside BASIC* on disk**

...and other related BASIC programming information and tools.

✓ **75 MB of Special Demos relevant for developers**

MacTech CD-ROM, Volumes 1-10:

$49 plus shipping and handling. $39 plus shipping and handling for upgrades from any previous version of the CD.

TCP/IP Scripting Addition

The Internet Scripting Solution

TCP/IP Scripting Addition v1.2 allows you to quickly develop Internet client/server applications. It grants access to MacTCP™ 2.0.4 and Open Transport™ by adding commands to AppleScript™.

- Supports ScriptEditor, FaceSpan™, and HyperCard™
- Build net–wise WebSTAR™ CGI scripts and NetScape™ CCI scripts
- Sample scripts include FTP, Gopher, Telnet, Post Office, E–Mail and more
- Featured on the Apple® Internet Server

ONLY $49

Order now through the MacTech Mail Order Store at 805–494–9797 or custservice@xplain.com

Mango Tree Software, Inc.
Box 1057 • Brookline, Massachusetts 02146
617–327–8663
http://www.mangotree.com/biz/mango

All trademarks are properties of their respective holders. Contact Mango Tree Software for site licensing information. Copyright © 1995 MTS, Inc.

```
class FW_CLASS_ATTR CBeeperFrame : public FW_CFrame,
                    public FW_MReceiver
{
//----------------------------------------------------------
// Initialization/Destruction
//
public:
  CBeeperFrame(Environment* ev, ODFrame* frame,
    FW_CPresentation* presentation, CBeeperPart* part);
  virtual ~CBeeperFrame();

//----------------------------------------------------------
// FW_CFrame overrides
//
public:
  virtual void CreateGadgetLayout(Environment* ev,
            FW_CGadgetInitializer& initializer);

  virtual ODDragResult CanAcceptDrop(Environment* ev,
            ODDragItemIterator* dragInfo);

  virtual void Draw(Environment* ev,
          ODFacet* facet, ODShape* invalidShape);

  virtual void FrameShapeChanged(Environment* ev);

//----------------------------------------------------------
// FW_MReceiver overrides
//
public:
  virtual void HandleNotification(
          const FW_CNotification& notification);

//----------------------------------------------------------
// Data Members
//
private:
  CBeeperPart* fBeeperPart;
  FW_CPushButton* fBeeper;
  ODID     fButtonId;
  FW_CHandleFunctionConnection fConnection;
  ODTypeToken   fButtonNotificationToken;
};
```

CBeeperFrame includes a button gadget, so it overrides the method CreateGadgetLayout to create the button gadget. CBeeperFrame can accept a drop of either scripts or sounds, so it overrides CanAcceptDrop. It overrides Draw to render itself, and it overrides the FrameShapeChanged notification to update its layout when the frame is resized. Finally, it overrides the HandleNotification method from the mixin class FW_MReceiver so that it can respond to button presses. Let's now look at the implementation of these methods.

CBeeperFrame::CBeeperFrame

```
CBeeperFrame::CBeeperFrame(Environment* ev,
          ODFrame* frame,
          FW_CPresentation* presentation,
          CBeeperPart* part) :
  FW_CFrame(ev, frame, presentation, part),
  fBeeperPart(part),
  fBeeper(NULL),
  fButtonId(0),
  fConnection(this),
  fButtonNotificationToken(0)
{
  fConnection.Connect();
  this->AddToFocusSet(ev,part->GetClipboardFocusToken(ev));
  SetDroppable(ev, TRUE);
}
```

The constructor initializes the base class and data members, and then sets up the notification connection, prepares to receive data from the clipboard, and registers itself as being interested in receiving data through drag-and-drop.

CBeeperFrame::~CBeeperFrame

```
CBeeperFrame::~CBeeperFrame()
{
  FW_CInterest interest(fBeeper, fButtonNotificationToken);
  fConnection.RemoveInterest(interest);
}
```

The destructor removes the notification connection.

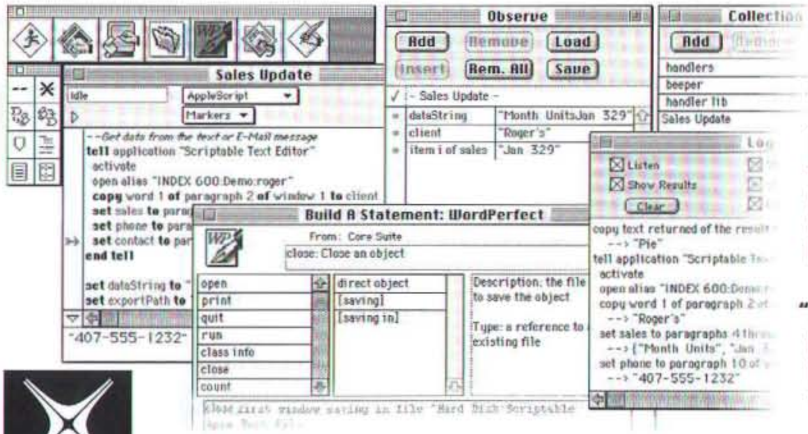
CBeeperFrame::CreateGadgetLayout

```
void CBeeperFrame::CreateGadgetLayout(Environment* ev,
            FW_CGadgetInitializer& initializer)
{
  FW_CDynamicString label("Beep");

  FW_CRect frameRect;
  this->GetFrameShapeBounds(ev, frameRect);

  fBeeper = FW_NEW(FW_CPushButton,
      (ev, initializer, this, fButtonId,
      frameRect[FW_kTopLeft], frameRect[FW_kBotRight],
      label));
  fBeeper->SetDefault(ev, false);

  fButtonNotificationToken =
      fBeeper->GetButtonPressedNotificationToken(ev);
  FW_CInterest interest(fBeeper, fButtonNotificationToken);
  fConnection.AddInterest(interest);
}
```

The #1 debugging and development environment for AppleScript™!

Scripter®

- **Real step-by-step debugging**
- **Variable watcher & expression evaluator: set variables, in context, during debugging!**
- **Enhanced trace log**
- **FaceSpan™ link**
- **Vocabulary access in a mouseclick**
- **Power-assisted statement construction**
- **Multifunction find & replace**
- **Automatic navigation to subroutines**
- **Background processing**



```
--Get data from the text or E-Mail message
tell application "Scriptable Text Editor"
  activate
  open alias "INDEX 600:Demo:roger"
  copy word 1 of paragraph 2 of window 1 to client
  set sales to parag
  set phone to para
  set contact to par
end tell

set dataString to "
set exportPath to "
"407-555-1232"
```

```
✓ - Sales Update -
= dataString        "Month UnitsJan 329"
= client            "Roger's"
= item i of sales   "Jan 329"
```

```
copy text returned of the result
  --> "Pie"
tell application "Scriptable Te
  activate
  open alias "INDEX 600:Demo
  copy word 1 of paragraph 2 of
  --> "Roger's"
  set sales to paragraphs 4 thr
  --> {"Month Units", "Jan 3
  set phone to paragraph 10 of
  --> "407-555-1232"
```

If you're an AppleScript novice, Scripter will take you by the hand and show you the correct syntax for your statements. If you know what you're doing, Scripter will help you do it faster than any other editor. Scripter lets you debug properly: truly line-by-line, so you catch the offending code in the act. And only in Scripter can you then fix the problem, and continue debugging — right from where you left off. In fact, Scripter has so many exclusive features, we can't list them all here. So get a copy today, and start scripting with power!

"Overall value ◆◆◆◆◆
Scripter 1.0 is top AppleScript editor and debugger... If you expect to do anything more than idle fiddling with AppleScript, you need Scripter." —*MacWeek, 6/12/95*

Main Event **1 800 616 8320** ORDERS ONLY +1 202 298 9595 mainevent@his.com AppleLink: MAIN.EVENT

CreateGadgetLayout creates a button of type FW_CPushButton, and then establishes its interest in receiving notifications for button clicks. The notification subsystem is a general-purpose mechanism for establishing event dependencies between objects. Objects may serve as Notifiers and Receivers. In our case, the push-button is a notifier, and the frame is the receiver. The FW_CInterest object establishes an interest in a particular kind of notification, in this case button pressed notifications. Adding the interest to the connection object establishes the connection. Once we establish the connection, button presses will result in the receiver's HandleNotification method (see below) being called.

CBeeperFrame::CanAcceptDrop

```
ODDragResult CBeeperFrame::CanAcceptDrop(Environment* ev,
               ODDragItemIterator* dragInfo)
{
  ODDragResult acceptDrop =
        FW_CFrame::CanAcceptDrop(ev, dragInfo);

  if (!acceptDrop)
  {
    for (ODStorageUnit *dragSU = dragInfo->First(ev);
               dragSU != NULL;
               dragSU = dragInfo->Next(ev))
    {
      if (CSoundAction::IsInStorage(ev, dragSU))
        return TRUE;
      if (CScriptAction::IsInStorage(ev, dragSU))
        return TRUE;
    }
  }

  return acceptDrop;
}
```

CanAcceptDrop iterates over the storage units of the OpenDoc drag item, looking for one that contains recognizable data. If it finds either a sound or a script, CanAcceptDrop returns true.

CBeeperFrame::Draw

```
void CBeeperFrame::Draw(Environment* ev,
           ODFacet* facet,
           ODShape* invalidShape)
{
  FW_CFacetContext fc(ev, facet, invalidShape);

  // Just erase and assume gadgets will draw
  FW_CRect invalidRect;
  fc.GetClipRect(invalidRect);
  FW_CRectShape::RenderRect(fc, invalidRect,
               FW_kFill, FW_kWhiteEraseInk);
}
```

The only rendering that a CBeeperFrame needs to explicitly do is to erase; the PushButton gadget will draw itself.

CBeeperFrame::HandleNotification

```
void CBeeperFrame::HandleNotification(
         const FW_CNotification& notification)
{
```

DragInstall® makes your installer choice *easy*

Easy for your users

DragInstall's unique drag-and-drop interface makes installing your software a piece of cake for your users. And cuts down on tech support calls.

Easy for you

DragInstall's Builder utility allows you to create complex installers in minutes not days. And without the need to learn a complicated scripting language.

Easy on your wallet

DragInstall's one-time fee of only $300.00 allows you to distribute an unlimited number of installers. No yearly renewals, no royalties, no hassle!

To find out just how easy DragInstall is, call us at
1-800-890-9880
to receive a free demo disk.
Or contact us at:

Ray Sauers Associates, Inc.
1187 Main Avenue, Suite 1B
Clifton, NJ 07011-2252 USA
Voice: 201-478-1970
Fax: 201-478-1513
E-mail: draginstall@sauers.com
http://www.sauers.com/draginstall

```
  const FW_CButtonPressedNotification& buttonPressed =
    (FW_CButtonPressedNotification&) notification;

  if (buttonPressed.GetButtonId() == fButtonId)
  {
    fBeeperPart->DoAction();
  }
}
```

This method is inherited from `FW_MReceiver`, and is called by the notification subsystem. This method first checks to ensure that it is receiving notification from the correct button, and then forwards the action request to the part (which will forward the request to the current `CAction` object).

CBeeperFrame::FrameShapeChanged

```
void CBeeperFrame::FrameShapeChanged(Environment* ev)
{
  FW_CFrame::FrameShapeChanged(ev);

  // Resize the Beeper to fit the new frame size
  FW_CRect frameRect;
  this->GetFrameShapeBounds(ev, frameRect);
  fBeeper->SetSize(ev, frameRect.BotRight());

  // Redraw the entire frame
  this->Invalidate(ev);
}
```

When the frame shape changes, the frame simply resizes the push-button to take up the entire frame.

CBeeperSelection

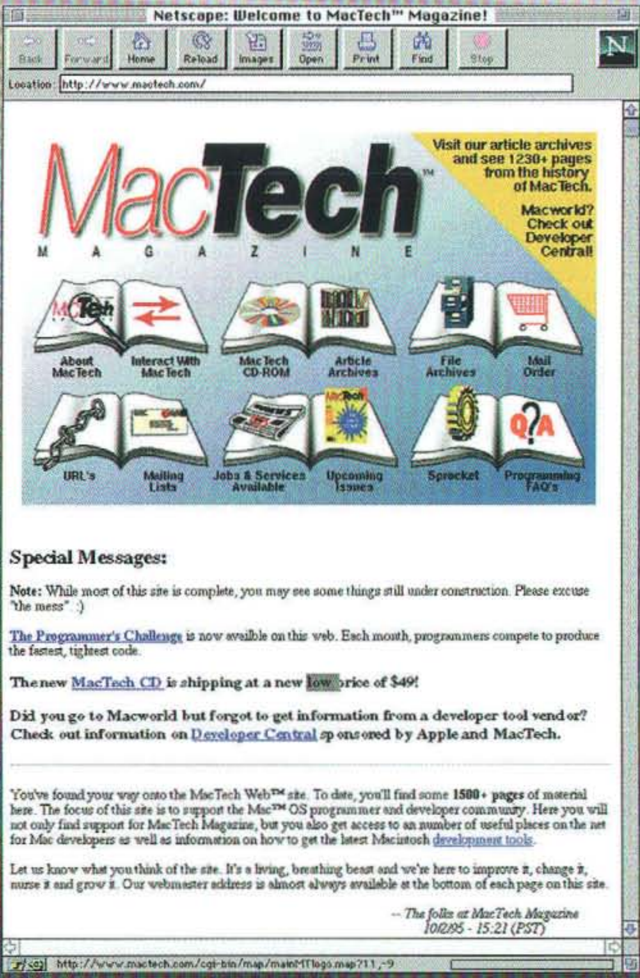
The ODF class `FW_CSelection` is an abstract base class used by ODF to represent the user's active selection of data. A Copy command will copy this data to the clipboard, drag operations copy or move the selected data, etc. `FW_CSelection` defines four pure virtual methods that must be overridden; `FW_CBeeperSelection` defines these and overrides two more. As it turns out, the "selection" of a Beeper part is always the entire data content, so the implementation of these methods is especially simple; most don't do anything.

```
class FW_CLASS_ATTR CBeeperPart;

class FW_CLASS_ATTR CBeeperSelection : public FW_CSelection
{
//----------------------------------------------------------
// Initialization/Destruction
//
public:
  CBeeperSelection(Environment* ev,
           CBeeperPart* beeperPart);
  virtual ~CBeeperSelection();

//----------------------------------------------------------
// Inherited API
//
public:
  virtual void CloseSelection(Environment* ev);
  virtual FW_Boolean ClearSelection(Environment* ev,
              FW_ClearSelection clearAfter);
  virtual FW_Boolean IsEmpty(Environment* ev) const;
  virtual void SelectAll(Environment* ev);

  virtual void DoExternalizeSelection(Environment* ev,
              ODStorageUnit* storage,
              FW_CCloneInfo* cloneInfo);

  virtual FW_Boolean DoInternalizeSelection(Environment* ev,
              ODStorageUnit* storage,
              FW_CCloneInfo* cloneInfo);

//----------------------------------------------------------
// Data Members
//
private:
  CBeeperPart* fBeeperPart;
};
```

CBeeperSelection::CBeeperSelection

```
CBeeperSelection::CBeeperSelection(Environment* ev,
                CBeeperPart* beeperPart) :
  FW_CSelection(ev, FALSE, FALSE),
  fBeeperPart(beeperPart)
{
}
```

http://www.mactech.com


Netscape: Welcome to MacTech™ Magazine!
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
N
Location: http://www.mactech.com/
MacTech™
M A G A Z I N E
Visit our article archives and see 1230+ pages from the history of MacTech.
Macworld? Check out Developer Central!
About MacTech
Interact With MacTech
MacTech CD-ROM
Article Archives
File Archives
Mail Order
URL's
Mailing Lists
Jobs & Services Available
Upcoming Issues
Sprocket
Programming FAQ's
http://www.mactech.com/cgi-bin/map/mainMTlogo.map?11,-9

Special Messages:

Note: While most of this site is complete, you may see some things still under construction. Please excuse "the mess" :)

The Programmer's Challenge is now availble on this web. Each month, programmers compete to produce the fastest, tightest code.

The new MacTech CD is shipping at a new low price of $49!

Did you go to Macworld but forgot to get information from a developer tool vendor? Check out information on Developer Central sponsored by Apple and MacTech.

You've found your way onto the MacTech Web™ site. To date, you'll find some **1500+ pages** of material here. The focus of this site is to support the Mac™ OS programmer and developer community. Here you will not only find support for MacTech Magazine, but you also get access to an number of useful places on the net for Mac developers as well as information on how to get the latest Macintosh development tools.

Let us know what you think of the site. It's a living, breathing beast and we're here to improve it, change it, nurse it and grow it. Our webmaster address is almost always available at the bottom of each page on this site.

-- The folks at MacTech Magazine
10/2/95 - 15:21 (PST)

(Need we say more?)

For Macintosh
Programmers & Developers

MacTech™
M A G A Z I N E

Guide Composer

Previewed at WWDC '95

Develop Apple Guide files for any new or existing application

- No Programming
- WYSIWYG Content Entry
- Coach Marks
- Index Terms
- PICTs
- Sounds
- Integrated with Guide Maker Lite (included) For Easy Compiles
- Optional Source Code Output
- Change back from your $100

Available from the MacTech Mail Order Store

Demo Available:
AppleLink, AOL, CIS,
eWorld, Internet:Info-Mac

MacOS

$99

STEPUP SOFTWARE

7110 Glendora Avenue
Dallas, Texas 75230
214-360-9301
214-360-0127 fax

Guide Composer is a trademark of StepUp Software. Guide Maker is a trademark of Apple Computer, Inc.

The CBeeperSelection constructor simply initializes its base classes and caches away a pointer to the part.

CBeeperSelection::~CBeeperSelection

```
CBeeperSelection::~CBeeperSelection()
{
}
```

The destructor doesn't need to do anything.

CBeeperSelection::CloseSelection

```
void CBeeperSelection::CloseSelection(Environment* ev)
{
}
```

In our case, closing a selection is a no-op...

CBeeperSelection::ClearSelection

```
FW_Boolean CBeeperSelection::ClearSelection(Environment* ev,
              FW_ClearSelection clearAfter)
{
  return FALSE; // Do nothing
}
```

...as is clearing the selection.

CBeeperSelection::IsEmpty

```
FW_Boolean CBeeperSelection::IsEmpty(Environment* ev) const
{
  return (fBeeperPart->GetAction() == NULL);
}
```

A CBeeperSelection is empty if it does not have an action.

CBeeperSelection::SelectAll

```
void CBeeperSelection::SelectAll(Environment* ev)
{
}
```

The Beeper's data is always selected, so there's nothing to do.

CBeeperSelection::DoExternalizeSelection

```
void CBeeperSelection::DoExternalizeSelection(
                Environment* ev,
                ODStorageUnit* storage,
                FW_CCloneInfo* cloneInfo)
{
  CAction *action = fBeeperPart->GetAction();
  if (action)
  {
    storage->AddProperty(ev, kODPropContents);
    action->Externalize(ev, storage);
  }
}
```

DoExternalizeSelection isn't a no-op, but it is still trivial; We merely forward the operation onto the action object.

CBeeperSelection::DoInternalizeSelection

```
FW_Boolean CBeeperSelection::DoInternalizeSelection(
                Environment* ev,
                ODStorageUnit* storage,
                FW_CCloneInfo* cloneInfo)
{
  FW_Boolean internalized = FALSE;

  CAction *action = NULL;
  if (CScriptAction::IsInStorage(ev, storage))
    action = new CScriptAction();
  else if (CSoundAction::IsInStorage(ev, storage))
    action = new CSoundAction();

  if (action != NULL)
  {
    action->Internalize(ev, storage);
    fBeeperPart->SetAction(action);
    fBeeperPart->Changed(ev);
    internalized = true;
  }

  return internalized;
}
```

DoInternalizeSelection is slightly more complicated. It should look familiar, since there is not much difference between this method and the `CBeeperPart::InternalizeContent` method. We simply determine type of data to internalize, create the appropriate action object, initialize it with the data, and then ask the part to make it be the current action object.

What have we done?

You've now seen all the custom code needed to create our enhanced Beeper part. It is not really a complete part yet; it is missing some obvious features that a truly useful Beeper should have, not the least of which is a better name!

One key feature that the Beeper needs is the ability to change the title of the button. If you look back over the code, you'll see that the button title is hard-coded to be `Beep` inside the method `CBeeperFrame::CreateGadgetLayout`. A real part would allow the default title to be specified from a resource, and provide some simple means of changing the title. One possibility is to allow dropping a text clipping as the means of setting the title. Another possibility is to set the button title to the name of the file dropped onto the button. No doubt you can think of other features that a button-as-part should have, such as a visual style more interesting than the simple rounded-rectangle.

If you look past these shortcomings, I believe it's fair to say that this example is a pretty impressive demonstration of the power of OpenDoc and ODF. With only a small amount of code, we have created a *truly reusable* software component. I'm not talking about the code-level reuse that we were able to achieve by using ODF, which is no small matter. I'm instead talking of the much more dramatic reuse achieved by high-level componentization. The `ODFBeeper` part is a component that can be reused without needing a single additional line of C++ code. End users with just a modest understanding of AppleScript could use the `ODFBeeper` component to create applications customized to their needs.

Clearly, the `ODFBeeper` component by itself, or even combined with a container part like `ODFDraw`, would not significantly empower end-users. We need quite a few more components before we'll see such a dramatic shift. However, considering how easy it was to create this component, it doesn't take much imagination to believe that we could soon have a wide assortment of powerful components.

Wrapping Up

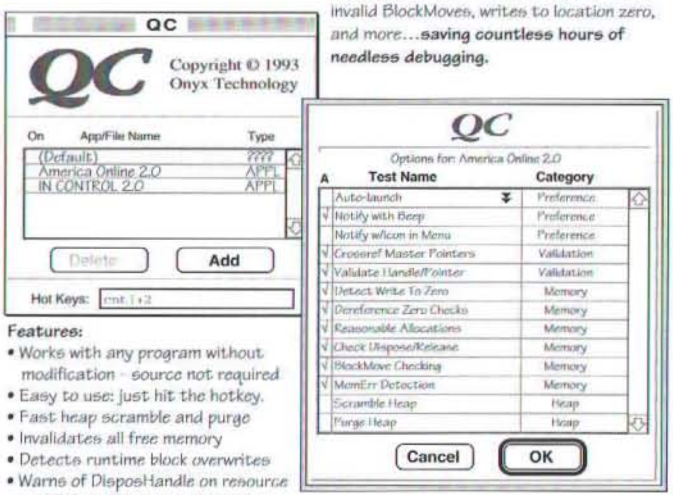
With that uplifting vision of the future, it's time for me to tie a bow around this article and ship it. One of my goals for this article was to give you a taste of what it's like to develop for OpenDoc with ODF. Hopefully it's left you thirsting for more. If you'd like more information about ODF, send a note to odfseed@apple.com; or feel free to write to me.

QC

NOW SHIPPING

QC: the Macintosh Testing solution. Subject your code to **brutal** stress conditions to **make it break** consistantly.

Or use QC during the development cycle to casually detect block boundary overwrites, invalid BlockMoves, writes to location zero, and more...**saving countless hours of needless debugging.**



QC

QC Copyright © 1993 Onyx Technology

On	App/File Name	Type
	(Default)	????
	America Online 2.0	APPL
	IN CONTROL 2.0	APPL

Delete Add

Hot Keys: cmt 1+2

QC

Options for: America Online 2.0

A	Test Name	Category
	Auto-launch	Preference
√	Notify with Beep	Preference
	Notify w/Icon in Menu	Preference
√	Crossref Master Pointers	Validation
√	Validate Handle/Pointer	Validation
√	Detect Write To Zero	Memory
√	Dereference Zero Checks	Memory
√	Reasonable Allocations	Memory
√	Check Dispose/Release	Memory
√	BlockMove Checking	Memory
√	MemErr Detection	Memory
	Scramble Heap	Heap
	Purge Heap	Heap

Cancel OK

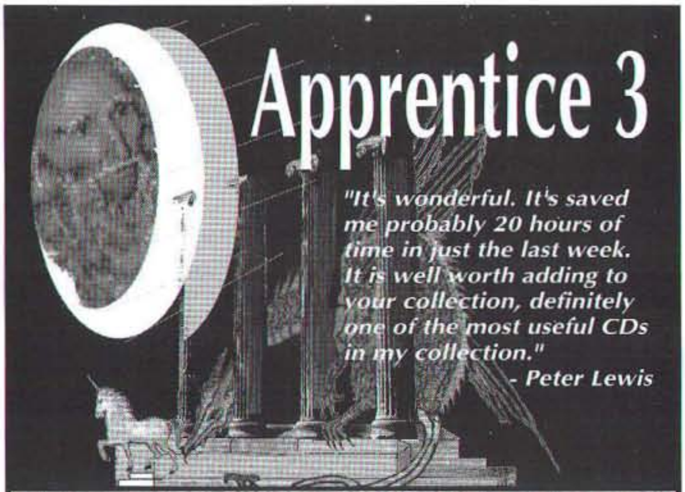
Features:
- Works with any program without modification - source not required
- Easy to use: just hit the hotkey.
- Fast heap scramble and purge
- Invalidates all free memory
- Detects runtime block overwrites
- Warns of DisposHandle on resource and ReleaseResource on handles.
- Validates BlockMove destinations
- Sophisticated heap verification
- Powerful API for precision control

$99 Special Introductory Price

30 Day
No Questions
Asked
Money Back
Guarantee

Onyx Technology
7811 27th Ave W.
Bradenton, FL 34209

Tel: 813.795.7801 Fax: 813.792.5152 AOL: OnyxTech ALink: D2238 CIS: 70550,1377


Apprentice 3
"It's wonderful. It's saved me probably 20 hours of time in just the last week. It is well worth adding to your collection, definitely one of the most useful CDs in my collection."
- Peter Lewis

Mac Source Code CD-ROM

If you're looking for great source code examples, this is the CD-ROM for you. Over 640 megabytes of high-quality and up-to-date Mac-only source code from hundreds of programmers. Over 80 percent of the examples and utilities are new to this release, mostly in C, C++, and Pascal, using CodeWarrior, Symantec, and MPW. Includes examples of applications, games, code resources, control panels, system extensions, plug-in modules, hundreds of snippets, and much more.

$35. Upgrade from any previous Apprentice release for only $25!

Shipping included for U.S. and Canadian orders. Add $5 for shipping outside the U.S. and Canada.

Visa, MasterCard, American Express, and Discover gladly accepted

Celestin Company, Inc., 1152 Hastings Avenue, Port Townsend, WA 98368
800 835 5514 • 360 385 3767 • 360 385 3586 fax
Internet: celestin@celestin.com • http://www.celestin.com/

DIALOG BOX

By Neil Ticktin, Editor-in-Chief/Publisher

Dylan Lives

Jim Gagnon [Dialog Box, September] may have many reasons for not using Dylan. However, "...unreasonable to ... expect success from a proprietary language..." shouldn't be one of them. [The fact that it isn't out yet in ready-for-prime-time form is a reason.]

The language spec is in the public domain. The keepers are at CMU...take a look at

http://legend.gwydion.cs.cmu.edu:8001/dylan/

The name Dylan remains with Apple for the purpose of controlling its usage to conforming implementations (aging entertainers who themselves appropriated the name notwithstanding). Apple may someday have a Dylan implementation available for Mac developers. Meanwhile Dylan-like things are "all over" on other platforms (and Macintosh). They aren't called Dylan because they aren't yet (as in "Mindy"...Mindy Is Not Dylan Yet).

– John W. Baxter, Port Ludlow, WA

CGIs in C – Some Tweaks

Just a few notes on the article "Writing CGI Applications in C" in the September, 1995 issue of MacTech.

POST arguments can be up to 32k in size (not 24 as indicated on pg. 36). The username and password are both limited to 32 bytes.

You state on pg. 41 that "you can not allocate or release memory within a thread context." This only applies to pre-emptive threads, which were only implemented in the 68K Thread Manager. Cooperative threads can do what the main application thread can do.

I haven't looked at the responder code recently, but it used to be missing the call to MaxApplZone() during startup. That becomes quite necessary if the application is going to be dealing with large chunks of data.

Thanks for writing the article – I got some useful ideas from reading it!

– Grant Neufeld

According to John O'Fallon, the point about using memory within a thread context is true, but remeber that a call to MaxApplZone is required before threads allocate memory. See MacTech, November 1994, P. 68 for more details. There are many things that Responder doesn't do, mainly because Responder is a simplified application to teach CGI programming. – Ed. nst

What Ever Happened to FORTRAN?

First, let me laud your publishing of articles on Prograph CPX. This certainly is the way of the future of programming on the Mac. Indeed it is a natural extension of the Mac's iconic interface. I'm also delighted to find that there are two books in the offing concerning the programming of CPX. My background has been in the procedural languages, and I need some extra help in making the culture change to OOP.

Second, let me moan. In the past MacTutor and then MacTech gave coverage to the language I use most of the time - FORTRAN (LS FORTRAN). But the past year or so has seen a withering and then total absence of coverage of this language - first Jorg's Folder, then Jorg, then FORTRANs from the Mail Order. I know it isn't the most modern language, but there is a considerable opus of large scientific/engineering codes written in FORTRAN which have been debugged over decades. Those who use these codes are first of all rather intimidated by the idea of converting them to other more modern languages due to their size and second do not relish the large amount of time it will take to make the new codes as reliable as the existing FORTRAN codes. So, I don't think that FORTRAN is dead by any means and it is disappointing not to have Jorg's Folder any more, especially with the evolution of the Power Mac versions of FORTRAN.

– Stu Cook, Mt St Hilaire, Quebec

[Stu: Our biggest reason for not running FORTRAN articles is that we've not seen anything worthy of printing. While FORTRAN articles shouldn't get the same volume of coverage that C++ or even Pascal should (since there are more users of those languages), it does deserve coverage. We're open to publishing FORTRAN articles – anyone interested in writing one? – Ed. nst]

But, You Missed The Point

Thank you for your comments at the end of my letter complaining about Symantec's C++ 8.0, but I think you missed the point of my last comment.

If it would be so hard to write a PowerPC version of THINK Pascal, then how about a compiler that will accept THINK Pascal source code and produce native PowerPC object code. We programmers could do our program development using the present THINK Pascal development environment in the 680x0 emulator.

The THINK Pascal 4.0.2 compiler in the emulator is so much faster than any native C or C++ compiler that there is no real speed penalty for using the present compiler. After we have a working 680x0 program, we can feed the source code to the PowerPC compiler to produce the final code.

Since we will only be using this compiler a few times per program development cycle, it can be a slow dog like the Symantec C and C++ compilers. It doesn't have to have an editor and debugger like a real development environment. It just has to reliably compile previously proven code that was developed using the present excellent THINK Pascal development environment.

– Fred Johnson, Knoxville, Illinois

[Fred, your prayers have (or will) be answered shortly. Will Iverson at Symantec tells us that Symantec has come to an agreement with Language Systems Corp. LS will provide an Object Pascal compiler that will drop into the Symantec Project Manager environment that should compile THINK Pascal 4.0.2 source code. For more information, contact Language Systems can be reached at http://www.langsys.com/langsys or 800-252-6479.

In addition, there is a version of the Universal Headers modified to work with THINK Pascal that Symantec will make available by the end of October, 1995. These files will be available on the next SDA CD and probably online. Check the MacTech web site (http://www.mactech.com/special.html) or the Symantec site for more information – Ed. nst]

[Many thanks to:

Daniel Lidoff, Atascadero, California
JTenor11@aol.com
S.A. Klein, Madison, Wisconsin
Anthony Panos
Luman Wing
Chris F. Riley, Miami, Florida

who also expressed similar opinions about wanting to bring THINK Pascal to Power Macintosh. – Ed. nst]

Brain Damaged Languages?

Fred Johnson hit the nail on the head with his letter, printed in the Dialog Box section of the March issue of MacTech. He praised the editor and linker of THINK Pascal, as compared to those of Symantec C++. But he did not go far enough [to fully expose the languages behind the environments].

I, like Johnson, am somewhat of a Mac hacker. I have programmed the Mac professionally, but I consider myself to be an engineer who programs rather than a programmer. I have programmed using THINK Pascal, THINK C, and Microsoft QuickBASIC.

The BEST feature about THINK Pascal is its debugger, LightsBug. It is superb and clearly better than anything else I have used. (Why hasn't Symantec made the C/C++ debugger more like LightsBug?) I can examine all global and local variables immediately, without having to first select them for display as in the THINK C debugger. I can see the chain of calling routines, examine register variables, look through the heap, modify variables, and set watch points. And I can also examine and set memory directly.

I also share Johnson's opinion of C and C++ (or rather, Symantec's "C with objects", which is as close as I have gotten to C++.) Both are a mess. For example, what kind of brain-dead language knows the type of a variable in a "printf()" function call, yet makes the programmer specify it again with a formatting control code? I end up debugging my "printf()" function calls. (And that is just one example. I could go on and on.)

I made two serious efforts to learn "C with objects" but gave up in frustration both times. Object Pascal was a breeze after wrestling with C. (Even the manuals bear this out: the "Object Pascal" chapter is 14 pages for version 4 of Pascal; the "Using Objects in Think C" chapter is 37 pages for version 5 of C.)

The Object Pascal implementation of object-oriented programming is simple and elegant. With the use of just 3 additional keywords ("object", "inherited", and "override") and one function ("member()"), I can program in an OOP language. Although I can't speak from experience with C++, if my experience with "C with objects" is any indication, C++ is a mess.

The world seems to have run after C, but after using both, I don't know why. C is best described as a cryptic high-level assembler. Pascal is a well-designed high-level language. C'mon Symantec, support Pascal.

– Steve Ross, steve_ross@smtp.svl.trw.com

Criticism of Programming Style

I just received the winning solution to the programmers problem. It uses several goto statements and is not object oriented. As the only programming journal for the Mac it would be nice if you'd adopt a more "progressive" approach to teaching programming. Old notions of "efficiency" are passe.

– Don Winston

Chasing Down Semper.Fi

Due to listserver changes, the instructions to subscribe to the Semper.Fi mailing list in the September Viewpoint were obsolete soon after we went to press. Here are updated instructions: email to listproc@abs.apple.com a message with this command in the body: "subscribe semper.fi <your name here>".

If you want to see what other lists are available from this listserver, send the "list" command, or see the web page http://abs.apple.com/apple-internet/.

UNIFORM RESOURCE LOCATORS

By Jim Straus, URLs@mactech.com

Spare your fingers and find this column online at:
http://www.mactech.com/URLs.html

Also, for a limited time, you can send mail to: **MacTech-URLs@class.com** and you'll receive the latest list back.

Latest Updates

Internet Related

List of Mac Mailing Lists	http://www.macfaq.com/mailinglists.html

New Technologies

Open Transport and PCI	http://pilot.njin.net/~msproul/

Other Programmer Resources

Fortran to C converter	ftp://alumni.caltech.edu/pub/Mac_F2C/
Grant's CGI framework in C	http://arpp1.carleton.ca/grant/mac/grantscgi/
MacHack	http://www.machack.com/
Parse CGI OSAX	http://marquis.tiac.net/software/home.html
PreFab Software	http://www.tiac.net/prefab/

Vendors, Products and Miscellaneous

Absoft	http://www.absoft.com/
MacHax Group	http://www.hax.com/hax.html
Celestin	http://www.celestin.com/
LPA Prolog	http://www.lpa.co.uk/
MacMagazine	http://oeonline.com:80/~tomk/MacMagazine/
MacNet Journal	http://www.dgr.com/web_mnj/
MacWeek	http://www.zdnet.com/~macweek/
The Ultimate Mac	http://www.freepress.com/myee/ultimate_mac.html
VIDI	http://erehwon.caltech.edu/vidi/vidi-homepage.html

Internet

Mailing lists are one of the more popular ways of distributing information on the Internet. They can reach a wider audience than news groups or the world wide web, be more selective in their distribution, and reach their readers in a more timely fashion. The Well Connected Mac has a list of over 150 Macintosh specific mailing lists that cover a large range of topics. You can subscribe to as many as your patience and mailbox will allow.
List of Macintosh Mailing Lists http://www.macfaq.com/mailinglists.html

Macintosh

There are several on-line magazines for the Macintosh community (besides MacTech, of course). You probably already know about MacWeek, but did you know that it is available on-line? In case you don't know about MacWeek, it is a weekly magazine that was often referred to as the way that one hand at Apple knows what the other hand was doing, and there is some truth to the joke. You can find it at:
MacWeek http://www.zdnet.com/~macweek/

Another new on-line publication is MacMagazine. It's first issue is September (when I'm writing this column), and the first issue has several good articles. I hope that by the time you read this it will have developed and the content will be even better. Check it out for yourself at:
MacMagazine http://oeonline.com:80/~tomk/MacMagazine/

A site that has been around for a while but is a good site to check out, is the ULTIMATE Macintosh. This site has links to tons of stuff. A good place from which to branch out into the Macintosh universe on the Internet. Branch out from:
The Ultimate Mac http://www.freepress.com/myee/ultimate_mac.html

Neat non-Macintosh sites of the Month

Keeping with the theme of electronic magazines, I would like to draw your attention to The Feed. It is a fully electronic magazine (no print counter-part) and is a view of what interactive publishing could be like in the future. Some very notable people contribute to The Feed and you can interact with them through The Feed. It is a professional magazine, well layed out, and with good content. Highly recommended at:
The Feed http://www.emedia.net/feed/

For those of you not in the Silicon Valley, you might want to check out the San Jose Mercury News. This is an on-line version a newspaper that covers the Silicon Valley. These are the same people that bring you Mercury Center on America On-line. The top stories and index are free, but there is a subscription charge to see everything (though it is a lot less than subscribing to the paper itself). They also have a service called NewsHound that can automatically scan the news for key words and send you the articles. A start at agent technology. Read the paper at:
Mercury Center http://www.sjmercury.com/

Well, that's it for this month. As aways, if you find something interesting, or have updates, send them to
URLs@MacTech.com

Thanks to, and many others for their contributions for their suggestions and pointers to new and old sites. Apologies to Jon Pugh for misspelling his name in the September issue!

Internet-Related Material

Apple-Internet mailing lists	http://abs.apple.com/apple-internet/
Bolo	http://bolo.ncsa.uiuc.edu/
Clay Basket	http://www.hotwired.com/staff/userland/clay/
Consensus	http://www.consensus.com:8300
CU-SeeMe	http://www.jungle.com/msattler/sci-tech/comp/CU-SeeMe/
DigiCash	http://www.digicash.com/ecash/ecash-home.html
ForeFront Group (GrabNet)	http://www.ffg.com/
Info-Mac Searcher	http://www.mid.net/INFO-MAC
Internet Config	ftp://ftp.share.com/pub/internet-configuration/
InterNIC	http://www.internic.net/

DataScript™

The Scriptable Database Client

DataScript is probably the **quickest**, **easiest** and most **cost-effective** way to make your integrated AppleScript solutions database aware - today.

quickest. You can make new or existing scripted solutions database aware in a flash. It takes just six lines of AppleScript to fetch data from an industry standard database management system such as Oracle, Sybase, DB2 or Informix. We think you'll be pleased with it's performance too - it launches in an instant and is available accelerated for PowerPC.

easiest. DataScript's scripting vocabulary or terminology is easy to learn, easy to use and easy to remember. If you've done any scripting before then you'll be running with DataScript in no time. If you're new to AppleScript don't worry - "Inside DataScript" contains lots of easy to follow script snippets which you can reuse in your own solutions.

cost-effective. Because DataScript is so easy to learn and easy to use you'll become productive very, very quickly. Once you're ready to ship your finished database aware solution you'll find our licensing schemes very attractive.

DataScript can be purchased from General Knowledge by phone, fax or INTERNET using VISA/Mastercard or American Express. DataScript is also available from the MacTech Mail Order Service

Telephone +44 1202 746 026
Fax +44 1202 715 600
INTERNET knowledgable@applelink.apple.com

ISP FAQ	http://www.amazing.com/internet/
ISDN page	http://alumni.caltech.edu/~dank/isdn/
Peter Lewis	http://www.share.com/peterlewis/
see also	ftp://ftp.share.com/pub/peterlewis/
see also	ftp://amug.org/pub/peterlewis
List of Mac Mailing Lists	**http://www.macfaq.com/mailinglists.html**
Lycos (Index of the Internet)	http://www.lycos.com/
MacHTTP (see also WWW)	
Mailing List	http://www.biap.com/machttp/mailing_list.html
Registry	http://www.batnet.com/ape/machttp_talk/machttpservers.by.mac.html
Extending MacHTTP	http://www.uwtc.washington.edu/Computing/WWW/Lessons/START_HERE.html
MacWeb	http://galaxy.einet.net/EINet/MacWeb/MacWebHome.html
Matthias Neeracher	http://err.ethz.ch/members/neeri.html
MERIT Searcher	http://pubweb.nexor.co.uk/public/mac/archive/welcome.htm
OpenTransport/TCP	gopher://seeding.apple.com
Outland	ftp://ftp.outland.com/
Portable Net. Graphics	http://sunsite.unc.edu/~boutell/png.html
ScriptWeb	http://www.gz.com/scriptweb/
Eric Scouten (TCP)	http://www.metrowerks.com/tcpip/index.html
SGML, Info	http://www.sil.org/sgml/sgml.html
C Parser	ftp://ftp.jclark.com/pub/sgmls
C++ Parser	ftp://ftp.jclark.com/pub/sp/
WWW, Jon W. Mac WWW Development Guide	http://www.uwtc.washington.edu/Computing/WWW/Mac/Directory.html
CGI demo/info	http://charlotte.acns.nwu.edu/mailtools/ ftp://acns.nwu.edu/pub/jlnstuff/mailtools/
Intro. to WWW	http://www.eit.com/web/www.guide/
Web66	http://web66.coled.umn.edu/
Non-Mac servers	http://www.w3.org/hypertext/WWW/Library/Status.html
Yahoo (Great index of the Internet)	http://www.yahoo.com/

New Technologies

AOCE	ftp://ftp.andrew.cmu.edu/pub/aoce/ http://www.contrib.andrew.cmu.edu/usr/jbbt/aoce/aoce.html
Apple, ftp	ftp://ftp.info.apple.com
Web interface to ftp	http://www.info.apple.com/cgi-bin/lister-pl
DTS	http://www.info.apple.com/dev/dts.html
Tech Notes	http://www.info.apple.com/dev/technotes/Main.html
see also	http://www.austin.apple.com
Apple Guide	http://www.guideworks.com/
Dylan	http://www.cambridge.apple.com/
see also	ftp://cambridge.apple.com/pub/dylan
see also	http://legend.gwydion.cs.cmu.edu:8001/dylan
see also	news:/comp.lang.dylan
Mailing list archive	ftp://cambridge.apple.com/pub/dylan/mail-archive/
Marlais	ftp://ftp.cis.ufl.edu:/pub/src/Marlais http://www.cis.ufl.edu/~jnw/Marlais/
Mindy	ftp://ftp.bdt.com//home/beard/Mindy-PPC.sit.bin
Kaleida	http://www.kaleida.com/
OpenDoc/Bento/SOM	http://www.cilabs.org/pub/cilabs/tech/ http://www.info.apple.com/dev/du/intro_to_opendoc/iod0_index.html
OpenDoc Part Ideas	http://www.eng.uci.edu/~sroussey/NetVision/software/od_parts/

Continued on page 62

NEWSBITS

By John Kawakami, MacTech Magazine Editorial Assistant

CodeWarrior 7 Delivers

The anticipated CodeWarrior 7 has shipped. CodeWarrior now comes on two CDs – Tools and Reference. They are designed so that you can install from the Tools CD, and pop in the Reference CD for, documentation and information.

The CodeWarrior IDE now has new Target preferences, with PlugIn preferences for languages (C/C++ and Pascal), target platforms (Mac 68K, CFM 68K, PPC, Win32/x86, and MagicCap), and tools to import from other formats, such as MPW libraries. The installer will install all the correct files for each platform you choose, including stationery.

Other changes include improved access paths, mod date checking, multibyte characters in comments & strings, Enable/Disable Debugging options toggle the Run/Debug menu item, debugging for non-applications, and searching in the Message window. The 68K and PowerPC debuggers have been merged; debuggers Win32/x86 and Magic Cap are still separate applications, but they all work in much the same way.

A new debugger nub, named Metronub also makes it's first public appearance on CW7. It is intended to replace DebuggerINIT and PowerMac Debug Services. Metronub is a fat binary on steriods: it supports 68K, CFM68K and PowerPC debugging services.

Changes in the C++ compiler: support for RTTI and new cast expressions, better error messages, new virtual function dispatch method, more efficiant class layout, improved code generation for inlines, support for throwing exceptions with destructors, multiple exceptions, new pragmas for import/export, and new reserved C++ keywords.

Warning: All C++ libraries and all precompiled header files must be regenerated for this version.

Pascal improvements: Pascal is now available as a compiler tool in the CodeWarrior IDE, for the Mac OS only. Includes new types: UNSIGNEDWORD (16 bits) and UNSIGNEDLONG (32 bits); new function: ReadString, which gets its input from a character string and can convert characters to different types; and PACKED ARRAY OF CHAR can now be mixed with the STRING type

68K improvements: Links all compiler-generated code and "unable to inline" code into segment 1, links very large projects faster.

PowerPC improvements: Smaller vtables, smaller and faster C++ code; new preference for storing static data in TOC, FMADD and FMSUB instructions, and traceback tables. Supports pipelining for 601, 603 and 604, new intrinsic functions, links very large projects faster, new Linker preference controls link memory/speed tradeoff, other optimization and code generation improvements. Makes source-level debugging info for inline and template functions when "Don't Inline" is selected. Allows you to throw exceptions from Shared Libraries.

Debugger improvements: the debugger is now a single, fat application that debugs 68K and PowerPC code in either C or Pascal. Now has support for debugging C++ exceptions and a menu item to turn this feature on or off. Also comes with a new debugger nub, MetroNub, to replace both DebuggerINIT and PowerMac DebugServices.

PowerPlant improvements: LString classes for Pascal strings improve string-handling; new array classes; and new pane classes: LIconPane, LGroupBox, LTextButton. In the Experimental folder: new table classes for multiple selection, variable-sized rows and columns, and different storage models.

Constructor improvements: Entirely rewritten in PowerPlant, contains Views, Text Traits and Custom Pane Types editors; many tools for editing PPobs including layout and hierarchy windows, automatic Tab and Radio Groups, new classes on Tools Palette.

In addition to the above, there are more improvements to other parts of the distribution. For more information, Metrowerks, email: info@metrowerks.com.

Fantasm Assembler

Programmers looking for a low cost assember will want to try Fantasm from Lightsoft. Ligthsoft claims "In tests against MPW. Fantasm was many times faster and produced code that was only a fraction of the size of that produced by MPW. Example: for a simple Macintosh editor based on te, MPW generated 82k, Fantasm generated 18k."

Version 3.20 of Fantasm was released recently, and all registered owners of 3.xx should have the latest version. If you are a registered user and have not received the update, send email to 100625.720@compuserve.com for more information.

If you are not a registered user of Fantasm there's a demo version, V3.20 (Unreg), available by ftp from:

ftp://ftp.demon.co.uk/pub/mac/fantasm

The OpenMac Free OS Developers Mailing List

There's a new mailing list dedicated to the technical aspects of porting different UNIX flavors, such as Linux and NetBSD (referred to here as the OpenMac projects), to the Macintosh and PowerMacintosh hardware.

Operating system developers need information about the specialized hardware and low-level driver software in the each Macintosh line. Until now, Apple has been slow to volunteer this information, and this mailing list attmempts to address this problem. Since the OpenMac projects are dedicated to the open exchange of information and source code, there is little likelihood of signing non-disclosure agreements.

The primary reason for this mailing list is to encourage discussion toward building better relations between the different OpenMac teams and those people interested in seeing the projects succeed. To this end, their goal is to maintain a list of specific information that can be presented to interested parties within Apple. Specifically, we encourage participation on the OpenMac homepage described below, and contribute your specific needs or solutions to the indicated problems.

The list maintainers emphasize that it's not a discussion group for feature requests or software support in the OpenMac OSs. For such information, browse the Web pages listed at the URLs below.

To subscribe to the OpenMac mailing list, you need to send mail to majordomo@puma.bevd.blacksburg.va.us with no subject and with body "subscribe openmac" on a single line. After your mail has been sent, you should receive an acknowledgement of your request, and short guide on using the listserv.

The primary online resource for the OpenMac ListServ is located at:

http://puma.bevd.blacksburg.va.us/openmac/

More information on the MacLinux or MacBSD projects in general is on the following pages:

http://nucleus.ibg.uu.se/maclinux
http://www.netbsd.org/Ports/mac68k/

(Another) PDA Developers Conference

Creative Digital Inc., publishers of PDA Developers magazine, is pleased to announce PDA Developers West '96, a technical conference designed to give developers that are creating PDA products:

- an in-depth technical understanding of the most popular hand-held platforms and their development environments;
- a broad overview of third-party PDA development tools and products;
- crucial information about what it takes to succeed as a PDA software developer;
- insight into wireless technologies and the service providers that are looking for development partners; and
- a glimpse at new, exciting technologies that will have an impact on the PDA market.

The first day of the conference consists of full-day tutorials on Magic Cap, GEOS, Newton, and Psion development. The second and third days include tracks on wireless development, advanced programming topics, the business of software development, and new technologies, plus keynotes by: Kheng Joo Khaw, general manager, handheld computer division, Hewlett Packard, Singapore; Andy Seybold, noted wireless industry analyst and editor of Andy Seybold's Outlook; and Jeff Hawkins, Chairman and founder of Palm Computing.

There will also be ongoing developer tools exhibits and demos all three days. To maximize attendee flexibility, there are separate registrations for the tutorials and the second and third days of the conference. PDA Developers West is scheduled for January 14-16, 1996, in San Mateo, CA, following the San Francisco MacWorld Expo. Individual tutorials are $150 until Nov. 30. Conference registration starts at $375 for both days, including breakfast and lunch. There are reduced rates for people who register for both a tutorial and the conference, student rates, and reduced rates for multiple attendees from the same company. Room rates at the Dunfey Hotel, where the conference will be held, start at $72 for a single or double room.

Further agenda and registration details can be found on: CompuServe (GO NEWTVENS, Creative Digital library), American Online (Computing/PDAs/PDA Development/New Files), eWorld (GO NEWTON/Llama Lounge/Sample Code/New Files), and ftp://newton.uiowa.edu/submissions.

Look for the files PDvWst96.sit and PDvWst96.zip, or contact Creative Digital Inc. for more information. PDA Developers West '96 is produced by Creative Digital Inc., and sponsored by General Magic, Inc. Creative Digital Inc., 293 Corbett Avenue, San Francisco, CA 94114, phone: (415) 621-4252, email: cdi@cdigital.com

Peregrine, a Prograph Client/Server Development Tool

Peregrine is a rapid applicaiton client/server development environment built on the power of Prograph CPX, a visual and object-oriented programming lnaugae for the Macintosh. In addition to a sophisticated development environment, Peregrine has a full featured internal SQL database engine for building standalone of client/server databases. Developers can create databases and applications without ever being connected to the target server.

Peregrine is offered in two versions. The Developer version is $1495 and supports Butler and dtF servers. The Corporate version is $2495 and includes connectivity to multi-user servers such as Oracle and Sybase as well. These prices include a subscription to email tech support and three CD ROMs containing upgrades and information. Pictorius Inc. (800) 927-4847

DEVELOPER
DU
UNIVERSITY

It's not just the basics anymore !

Advanced courses from Developer University
get you up to speed quickly on new Apple technologiess

❑ *OpenDoc*

❑ *PowerPC*

❑ *Newton*

❑ *Graphics/Imaging*

❑ *Apple Guide*

Courses Available as

Self-Paced Classroom Lecture Online

For more detailed information, check out our World Wide Web pages, http://www.info.apple.com, or contact the Apple Developer University Registrar at (408) 974-4897 or fax (408) 974-0544.

Developer University, Apple Computer, Inc. 1 Infinite Loop, MS 305-1TU, Cupertino, CA 95014

PowerTap™

The PowerTap API brings distributed computation to your application and makes it as easy as a simple black box. Divide your job into tasks, submit the tasks to the PowerTap API and then request task results.

Whether it's ray tracing, chemical design, video special effects or number-crunching in another field, your computations can most likely be done faster by mulitple machines.

- Easy-to-use API
- Robust
- Compatible
- Thread Manager Enabled
- Supports native, 68K and A4-based software.

Emerson Kennedy

(800) 297-3888
powertap@aol.com

PO Box 2530 • Redmond, WA 98073

OpenTransport/TCP	gopher://seeding.apple.com
	ftp://seeding.apple.com/ess/public/mactcp/MacTCP_Dev_Kit
Open Transport and PCI	**http://pilot.njin.net/~msproul/**
PowerPC	http://power.globalnews.com/
Quickdraw GX, Apple	http://www.info.apple.com/gx/gx.html
Fanclub	http://aah.residences.ulaval.ca/quickgx/quickgx.html
Lawrence D'Oliveiro	http://www2.waikato.ac.nz/ldo/gx/index.html
Quicktime	http://quicktime.apple.com/develop.html
see also	http://www.info.apple.com/dev/devinfo/quicktime/quicktime.html
Taligent	http://www.taligent.com/

Other Programmer Resources

4D	http://www.netf.org/4d/4d.html
ACIUS	ftp://ftp.dnai.com/pub/acius/
Ada	ftp://ftp.seas.gwu.edu/pub/ada
Animation Class Library	ftp://ftp.virtually.mcnet.ch/virtually/
Applescript	ftp://gaea.kgs.ukans.edu/applescript http://www.gz.com/scriptweb/
BMUG	http://www.bmug.org/
Brian Bechtel	http://www.info.apple.com/dev/geeks.html
Forth, MOPS,Yerk	http://www.netaxs.com/~jayfar/mops.html
MOPS	ftp: //taygeta.oc.nps.navy.mil/pub/Forth/Mops
Yerk	ftp: //astro.uchicago.edu/pub/MAC/Yerk
Fortran to C converter	**ftp://alumni.caltech.edu/pub/Mac_F2C/**
Free Development Systems Faq	http://www.europa.com/~sf/fdsm.html
Get1Resource	http://www.asel.udel.edu/~haynes/g1r.html
Grant's CGI framework in C	**http://arpp1.carleton.ca/grant/mac/grantscgi/**
Robert Lentz	http://www.astro.nwu.edu/lentz/mac/programming/home-prog.html
Lisp, MCL	http://www.digitool.com/
General	http://www.cs.rochester.edu/u/miller/alu.html
MacHack	http://www.machack.com/
MacTech	http://www.mactech.com/
Matthias Neeracher	http://err.ethz.ch/members/neeri.html
Matthew Xavier Mora	http://xavier.sri.com/umpg/umpg.html
nick.c (good for beginning Macintosh programmers!)	http://www.pitt.edu/~nick/
Parse CGI OSAX	**http://marquis.tiac.net/software/home.html**
François Pottier	http://acacia.ens.fr:8080/home/pottier/index.html
also	news://comp.sys.mac.digest
Digests archive	ftp://ftp.dartmouth.edu/pub/csmp-digest
PreFab Software	**http://www.tiac.net/prefab/**
Jon Pugh (AppleScript)	ftp://ftp.netcom.com/pub/jo/jonpugh/homepage.html http://www.infoworkshop.com/~jonpugh/default.html
Paul Robichaux	http://www.fairgate.com/paul.shtml
Source code, Apple	http://www.info.apple.com/dev/devinfo/macsourcecode.html
also Apple	http://www.info.apple.com/dev/source/index.html
also UMich	ftp://mac.archive.umich.edu/mac/development/source/
also M. Neeracher	ftp://ftp.switch.ch/software/mac/src/HTML/Welcome.html
Christopher Tate	http://world.std.com/~ctate/mac.html
TCL	ftp://daemon.ncsa.uiuc.edu/TCL/
see also	http://rhino.harvard.edu/dan/TCLArchive.html
TopSoft	http://www.topsoft.org/
XCMD Hideout	http://www.nmc.csulb.edu/brendon/xcmdhideout/xcmdhideout.html

Vendors, Products and Miscellaneous

Absoft	**http://www.absoft.com/**
Aladdin HASP	http://www.hasp.com
Aladdin Systems (Stuffit)	http://www.aladdinsys.com/
Alpha (text editor)	http://www.cs.umd.edu/~keleher/alpha.html
BBEdit	http://www.tiac.net/biz/bbsw/
MacHax Group	**http://www.hax.com/**
CastTech virtual castings	http://www.castech.fi/
Celestin	http://www.celestin.com/
Ceneca (PageMill)	http://www.ceneca.com/
Chris W. Johnson	http://gargravarr.cc.utexas.edu

CIL	http://www.cilabs.org/
Cyno Technologies	http://www.cyno.com/cyno
Digitool	http://www.digitool.com/
Dilbert	http://www.unitedmedia.com/comics/dilbert
DragInstall	http://www.sauers.com/draginstall
Tom Engel	http://gasnet.med.yale.edu/demos/Tom's_Macintosh_Page.html
Erogonomics Web	http://tucker.mech.utah.edu/
EduPage newsletter	mailto:listproc@educom.edu
in the body of the message put:	subscribe edupage your name
Evatac (Preditor)	http://www.access.digex.net/~evatac/
Graphical Business Interface	http://www.gbi.com/
General Magic	http://www.genmagic.com/
Global Village Communication	http://www.globalvillage.com/
Iconix	http://www.biap.com/Iconix/
Steve Jasik	http://www.jasik.com/
Just Some Guy	http://www.spies.com/greg/
Language Systems	http://www.langsys.com/langsys
FTP site	ftp://cais.com/pub/langsys
LPA Prolog	**http://www.lpa.co.uk/**
MIND/MacDNS	http://cybercronx.techwood.org
MacMagazine	**http://oeonline.com:80/~tomk/MacMagazine/**
Macintosh Vendor Directory	http://www.macfaq.com/vendor.html
MacLinQ	http://www.wsmith.com/maclinq/
MacNet Journal	**http://www.dgr.com/web_mnj/**
Mac Netswitch	http://www.nd.edu/~dwalton1/
MacNosy	ftp://ftp.netcom.com/pub/ma/macnosy
MacWeek	**http://www.zdnet.com/~macweek/**
Metrowerks	http://www.metrowerks.com/
Motorola	http://www.mot.com/PowerPC
NaviSoft	http://www.navisoft.com/index.htm
NeoLogic	http://www.neologic.com/~neologic/
Newton Source	http://www.newtonsource.com/
Newton	http://newton.uiowa.edu/
PACE	http://paceap.com/pace.html
Pictorius (Prograph)	http://192.219.29.95/home.html
QKS/Smalltalk	http://www.qks.com
QUED/M	ftp://ftp.nisus-soft.com/pub/nisus
see also	http://www.nisus-soft.com/~nisus
QuickCam	http://www.jungle.com/msattler/sci-tech/comp/hardware/quickcam.html
Rainbow Technologies	http://www.rnbo.com
RTFM Productions	http://rampages.onramp.net/~rtfm/
Semper Fi mailing list	mailto:listproc@abs.apple.com
in the body of the message put:	subscribe SEMPER.FI your name
StepUp Software	http://rampages.onramp.net/~stepup/
Summit Software	http://www.summsoft.com
Symantec	http://www.symantec.com/
Tenon (MachTen Unix)	http://www.tenon.com/
TidBITS newsletter	http://www.dartmouth.edu/pages/TidBITS/TidBITS.html
see also	news://comp.sys.mac.digest
to subscribe	mailto:info@tidbits.com
Time Tracker	http://www.maui.com/~billm
The Ultimate Mac	**http://www.freepress.com/myee/ultimate_mac.html**
UserLand AutoWeb	http://www.hotwired.com/Staff/userland/
Verb Finder	ftp://gaea.kgs.ukans.edu/frontier/oldstuff/VerbStack.sit.hqx
VIDI	**http://erehwon.caltech.edu/vidi/vidi-homepage.html**
The Well Connected Mac	http://www.macfaq.com/
Working Software	http://www.webcom.com/~working/OEM.html

Visit MacTech Magazine's Web site!
http://www.mactech.com

NEW VERSION **ScriptWizard**

Version 1.5

Now shipping. With all the new features that you asked for. ScriptWizard™ is the best-selling OSA/AppleScript script-editing and debugging tool that combines the power of a professional development environment with the ease of use that you expect of Macintosh® software. ScriptWizard improves your productivity by delivering testing and debugging facilities that are as intuitive as they are powerful, including a Variable Watcher, true single-step debugging, rapid script navigation, and more.

New features...

- **True AppleScript statement–level single stepping.**
- **Scriptable and Recordable.**
- **Improved text editing,** with drag and drop.
- **"Native"** for both **Power Macintosh** and **68K.**
- **Quick access** to **Dictionaries** from a new menu.
- **Full control** over **Toolbar** position & visibility.
- **Re-sizable panes** in windows.
- **Projector-aware** for source code control.
- **Edit and debug scripts for FaceSpan 2.0.**
- and much, much, more...

Upgrade $39

US price still only $99

US Orders: 800 232 1560
Tel: 510 843 6484. Also sold by APDA & MacTech.
Full Moon Software. Internet: 'sales@fullmoon.com'
International Orders: Tel +44 1628 660242. Fax 666084.

Can you spot the difference?

Plenty of people can't. Because whether you update software with a full set of program disks, or a file made with UpdateMaker 2, the result is the same. Guaranteed. UpdateMaker updates are totally ***reliable***. Its system of 32-bit checksums ensures that it updates the right file.

And UpdateMaker is ***easy-to-use*** – simply specify the files and UpdateMaker builds the update. There is no scripting or use of ResEdit. It's even easier for end-users – just one button to press.

UpdateMaker 2 works with any type of Macintosh file. The updater files are extremely ***compact***. And the program options numerous. You can preserve or override user customisations. Save files in Binhex format. Update up to 20 old versions with one file.

The real difference is the savings in time and money. Which explains why some of the best-known names in software development have already discovered UpdateMaker 2.

Accelerated for Power Macintosh

Distribution is unrestricted and royalty-free.
Only $225, order now, fax (415) 964 2886
Applelink:MacLab007 Internet:MacLab007@AppleLink.Apple.com

UpdateMaker 2™

ADInstruments, 2225 Grant Road, Suite #4, Los Altos, CA 94024 Phone: (415) 964 2878

MacRegistry™

Developer Job Opportunities

If you are a Macintosh developer, you should register with us! We have a database that enables us to let you know about job opportunities. When we are asked to do a search by a client company the database is the first place we go. There is no charge for registering. The database service is free. Geographic Coverage is nationwide.

Marketability Assessment - To get a specific feel for your marketability send a resumé via Email or call. You may also request a Resume Workbook & Career Planner.

Discreet - We are very careful to protect the confidentiality of a currently employed developer.

Scientific Placement is managed by graduate engineers, we enjoy a reputation for competent & professional job placement services and we are Mac fanatics.

1-800-231-5920 • das@spi.com • Fax 1-800-757-9003
http://www.scientific.com

Scientific Placement, Inc.

MT, Box 19949, Houston, TX 77224, 713-496-6100 Fax: 713-496-0373
MT, Box 71, San Ramon, CA 94583 510-733-6168 beth@spica.bdt.com
MT, Box 202676, Austin, TX 78720-2676 512-260-0123 lej@zilker.net
MT, Kenmore Station, Box 15225, Boston, MA 02215 617-424-8372 jen@spbos.pn.com
AppleLink: D1580; Compuserve: 71250,3001; eWorld: spi: AOL: davesmall

Work in Vacationland!

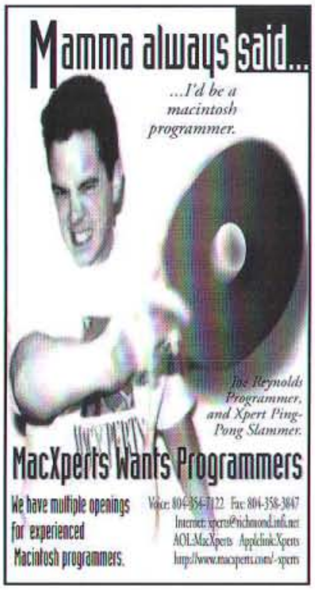
YOU DON'T HAVE TO LIVE IN THE URBAN JUNGLE TO DO WORLD-CLASS PROGRAMMING FOR EXCITING PRODUCTS!

A financially strong, rapidly growing software publisher, located in the heart of "America's Vacationland," is seeking strong C programmers with experience writing commercial products for Macintosh.

In addition to a competitive salary and benefits, and challenging, high-profile work, you'll enjoy the chance to live in a beautiful, low-cost community just a short drive from some of the nation's premier sporting areas and resorts. Live where most people only get to vacation.

To apply, send resume in confidence to:
Mark Swanson, Vice President, Software Engineering
Advantage Learning Systems, Inc.
P.O. Box 8036, Wisconsin Rapids, WI 54495-8036

ADVANTAGE
Learning Systems, Inc.



Mamma always said...

...I'd be a macintosh programmer.

Joe Reynolds Programmer, and Xpert Ping-Pong Slammer.

MacXperts Wants Programmers

We have multiple openings for experienced Macintosh programmers.

Voice: 804-354-7122 Fax: 804-358-3847
Internet: xperts@richmond.infi.net
AOL:MacXperts Applelink:Xperts
http://www.macxperts.com/~xperts

MacTech Magazine is your recruitment vehicle

When you need to fill important positions at your company, MacTech Magazine is the consistent choice of companies across the country for hiring the best qualified Macintosh programmers and developers. Let MacTech Magazine deliver your recruitment message to an audience of over 27,000 qualified computer professionals.

Call Ruth Subrin at 805/494-9797

The Trattner Network

The Trattner Network, a digital talent source, is looking for experienced Macintosh developers for a variety of consulting opportunities. Current projects include development in the areas of OpenDoc, Newton, Power PC, Metrowerks, PowerBook, MacApp, 4D and many others.

We have *urgent* needs for:

- Software Developers
- Hardware/Firmware Engineers
- QA/QC Professionals
- MultiMedia Developers
- Project Coordinator/Manager
- Network Professionals

The Trattner Network brings 10 years experience in the Macintosh consulting and placement industry, offering a unique blend of humanistic interaction and technical knowledge. Positions in Northern California and Nationwide.

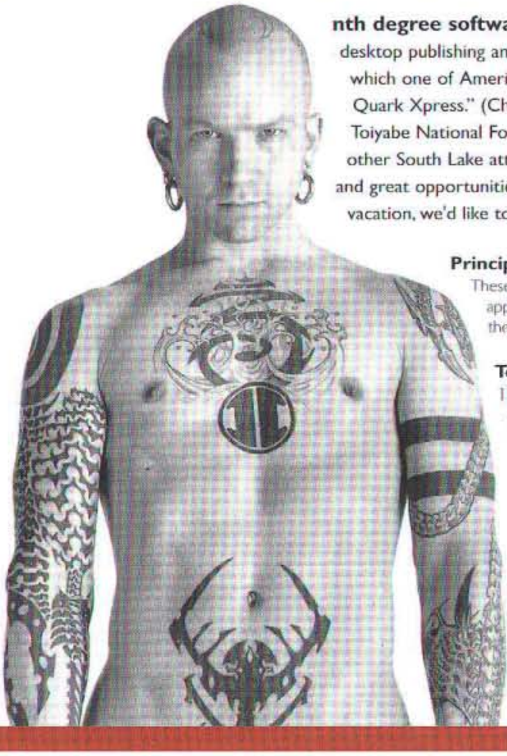
If you are looking for the chance to enhance your skills, team with the best, and make lots of money, send, fax or link your resume to:

TTN
Attn: Nyla Miller
170 State St., Suite 240, Los Altos, CA 94022
Phone (415) 949-9555 • Fax (415) 949-1026
AppleLink: trat.net • email: nmiller@tratnet.com

Do you have a service that you want to advertise to over 30,000 Macintosh programmers and developers?

Do it in the MacTech Magazine Classifieds –
an inexpensive vehicle that *gets results.*
Call Ruth Subrin at 805/494-9797 to reserve space NOW!

Career decisions can live with you forever.



nth degree software is a financially strong, rapidly growing Macintosh developer staking out a leading position in desktop publishing and multimedia enabling applications. We're the creators of Proteus, "The Magazine Architect", which one of America's largest publishing companies has called "the most brilliantly realized publishing system since Quark Xpress." (Check out the lead story in MacWeek's 9/18/95 GA section). Our headquarters is in the heart of the Toiyabe National Forest in income tax-free Nevada, just a short walk from Lake Tahoe and minutes from Heavenly and other South Lake attractions. If you're interested in an exciting work environment, an excellent compensation package, and great opportunities for personal growth in a beautiful, family-friendly community where most people only get to vacation, we'd like to hear from you immediately. Positions currently available include:

Principal Software Engineers • Senior Software Engineers • Software Engineers

These technically and analytically challenging positions, with responsibility for creating and enhancing high-profile applications, require object-oriented fluency and demonstrable experience developing commercial products in one of the following frameworks (in order of preference): MacApp 3.x, Metrowerks PowerPlant, or Think Class Library 2.x.

Technical Support Managers • Technical Support Analysts

These positions, with responsibility for end-user training and application support, require demonstrable experience in publication configuration and production, preferably at the Production Manager level or above. These positions also call for a high degree of familiarity with the use of System 7.5 and Macintosh publishing applications such as Quark Xpress, Pagemaker, and/or Photoshop. Some of these positions are available at our training facility in the New Jersey Meadowlands, just outside of New York City.

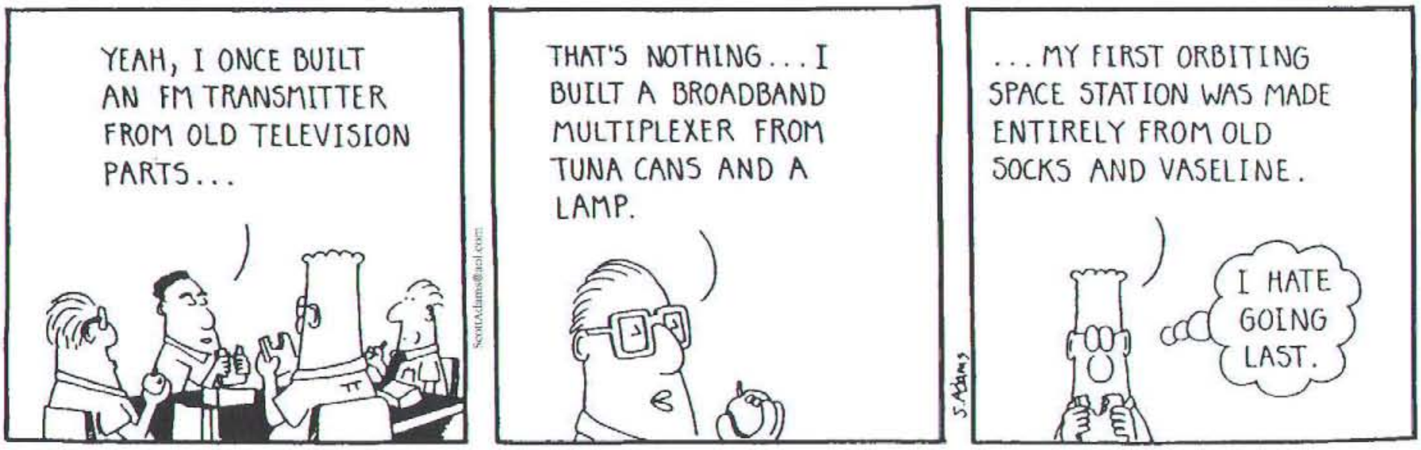
Regional Account Executives

These positions, with responsibility for initial client contact and account maintenance, require extensive experience in packaged software sales, ideally to clients in the publishing industry or related sectors.

Due to the immediacy of our staffing needs, we encourage you (or a friend) to send us your resume today, indicating the position of interest, by fax (702-588-4901) or e-mail (nthdegree@sierra.net). If you are driven to achieve, we'll take your career to the nth degree.

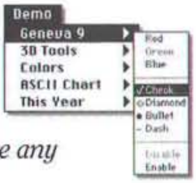
nth degree software

Dilbert® *by Scott Adams*


YEAH, I ONCE BUILT AN FM TRANSMITTER FROM OLD TELEVISION PARTS...
ScottAdams@aol.com
THAT'S NOTHING... I BUILT A BROADBAND MULTIPLEXER FROM TUNA CANS AND A LAMP.
S. Adams
... MY FIRST ORBITING SPACE STATION WAS MADE ENTIRELY FROM OLD SOCKS AND VASELINE.
I HATE GOING LAST.

MENU MILL

is a collection of five menu definition resources (MDEFs) that you just paste into your project's resource file.



Build a menu using the MDEF and then just handle menu selections like any normal menu.

Also contains Maitre d', a special menu develoment tool that assists you in designing your MENU MILL Menus.

MENU MILL

· Use your drawing programs as programming tools
· Fast unobtrusive RGB color picker
· Space saving Geneva 9 MDEF
· Calander Menu

JUST...$69.95

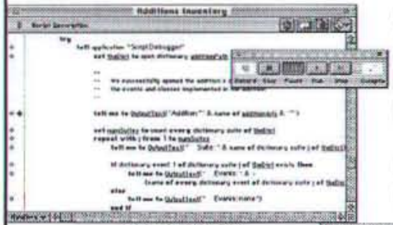
Write, Ariel Publishing, Inc.
11A Leisure Time Drive,
Diamondhead, MS 39525
Call (601) 255-6713
FAX (601) 255-7086

DEVELOPED BY STAZ SOFTWARE, INC

Late Night SOFTWARE

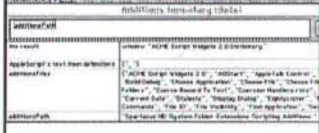
ScriptDebugger

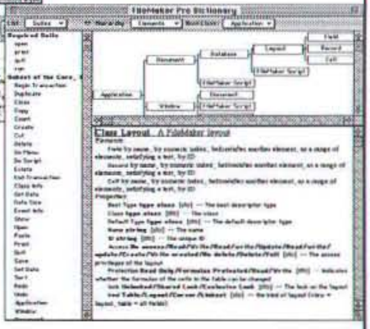
Script Debugger is a powerful and flexible AppleScript™ authoring tool that makes it easy for novice and experienced script writers to get the most from AppleScript.



Script Debugger

- Offers true single-step execution of all AppleScript scripts
- Provides a powerful scripting environment that includes Drag & Drop editing
- Is Power PC Native
- Is scriptable and attachable





What the Scripting Gurus Say About Script Debugger

"I'm in the midst of developing one of my largest scripts ever, and couldn't do it without Script Debugger. It now has 20 handlers, 15 properties and assorted globals. It's over 20K (of text alone) and is growing quickly. That's when you really appreciate Script Debugger."
Steve Michel, author and scripting authority

"I use Script Debugger as my AppleScript Script Editor and it is truly indispensable. I especially like the program's scriptability and attachability as these features make it infinitely customizable."
Jon Pugh, AppleScript pioneer, author of Jon's Commands

$129

Late Night Software Ltd.
Voice (604) 929-5578
Fax (604) 929-4961
E-mail gtubin@wimsey.com

Tips and Tidbits continued from page 80

Informative Cursors

One technique that I have used in the past where dropping into the debugger wasn't an option, and logging wasn't getting flushed in time/took too long, was to create a bunch of cursors numbering 00 - 99, and made a call to set the cursor and return the number of the previous cursor:

```
routine1()
{
short oldCursor = setDebugCursor(15);
    ...
    (void) setDebugCursor(oldCursor);
}
```

This way when the machine froze, the cursor would tell me what routine it had frozen in.

Tom Kimpton

MPW Editor Primitives

The MPW Command Reference describes 32 editor primitives which may be attached to any key sequence using the SetKey command. While the SetKey command itself is useful, the list of editor primitives alone is useful to MPW script writers.

Though only documented for use with SetKey, editor primitives may be used like any other MPW command: in scripts, command aliases, or AddMenu items. There are a couple of advantages:

1) They only work on the active window and have sharply defined functions, you don't have to remember any selection expressions or argument lists. Compare:

```
MoveStartOfFile
```

with

```
Find • "{Active}"
```

2) They're fast. I haven't put a stopwatch to it, but you can see the difference immediately in running a script that uses the primitives and one that doesn't.

The MPW Command reference lists these 32 primitives, the names of which should be self-explanatory:

DeleteCharLeft
DeleteCharRight
DeleteEndOfFile
DeleteEndOfLine
DeleteStartOfFile
DeleteStartOfLine

DeleteWordLeft
DeleteWordRight
MoveCharLeft
MoveCharRight
MoveEndOfLine
MoveLineDown
MoveLineUp
MovePageDown
MovePageUp
MoveStartOfFile
MoveStartOfLine
MoveWordLeft
MoveWordRight
SelectCharLeft
SelectCharRight
SelectEndOfFile
SelectEndOfLine
SelectLineDown
SelectLineUP
SelectPageDown
SelectPageUp
SelectStartOfFile
SelectStartOfLine
SelectWordLeft
SelectWordRight

There are also at least five primitives that aren't in the command reference:

ScrollEnd
ScrollHome
ScrollPageDn
ScrollPageUp
DebuggerCommand

I can understand why Apple might not want users to casually drop into MacsBug from an MPW script, but I wonder why they chose not to document the scrolling commands? Whatever the reason, you should obviously use caution when playing with undocumented features.

Lee David Rimar
Absoft Corporation

TestTrack™

Bug Tracking the Macintosh Way

TestTrack automates and simplifies tracking your bugs so you can concentrate on fixing them. Easy to set up, easy to use, and quick too!

- Track bugs, change requests, testers, configurations, and more
- Produce concise reports and charts
- Automatically route bugs to team members
- Minimum setup time
- Multiple users, full security

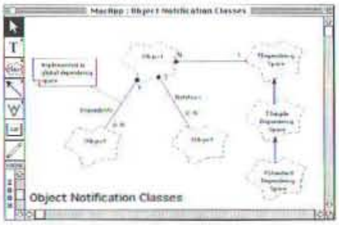
Available in September

For more information, call or send e-mail to seapine@one.net

Seapine Software

Seapine Software, Inc.
1066 Seapine Court
Maineville, OH 45039
513.683.6456

MultiQuest S-CASE™



The Clear Choice for Serious C++ Architects

Take control of your C++ development with S-CASE, a powerful design environment for today's demanding applications. Visualize system requirements using the Booch notation. Generate C++ code automatically from your design. Keep your models and code in perfect synchronization. Call for your ***free demo disk*** and find out why companies like Motorola, Siemens and Amoco rely on S-CASE.

Key Features:
- Booch notation (1994)
- Real time rule checking
- Iterative C++ code generation
- Hierarchical project manager
- Multi-user
- Multi-platform
- Check in / check out
- Color support

Benefits:
- Rapid prototyping
- Cleaner architectures
- Reusable designs

Platforms:
- Macintosh
- Windows
- Sun SPARC
- HP 9000

New! Version 2.0

Phone (708) 397-9930 **Fax (708) 397-9931**

Email 72531.2510@compuserve.com

Viewpoint continued from page 4

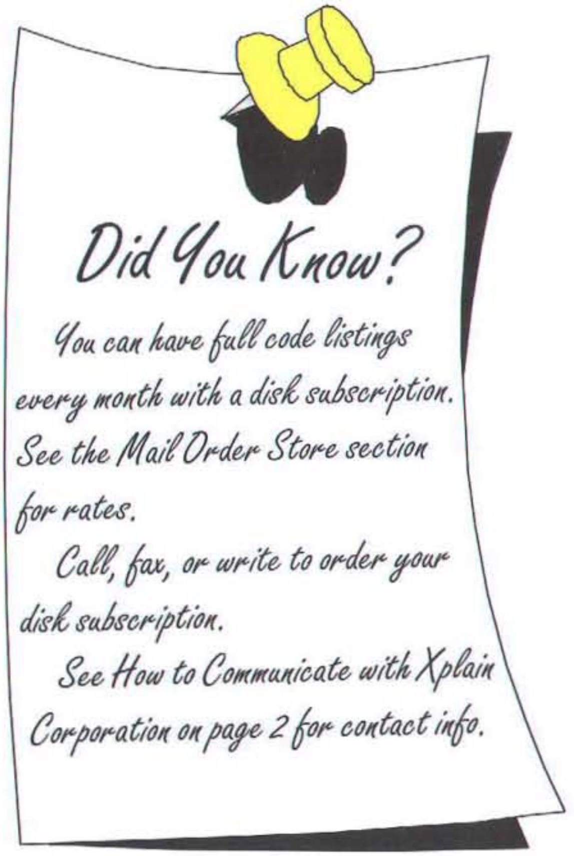
The Mac's a toy, right? Fine, ask them, "What kind of toy builds products like Adobe Photoshop? What kind of company uses Macintosh to build products which contribute to a $598M annual revenue cash flow?" Coal miners riding in coal cars ("Got a PC? You must be unenlightened."), dumb dads configuring a PC (how many millions of PC users are going to believe that they're really dumb if they have a PC that gets the job done?), and a crazed maniac taping an 800 number up on a glass wall (who knows what this means!?!). Could really smart developers who build tremendously great software and make a gazillion bucks be any worse?

Trivia Bits

Why is Ticks one of the worst low memory globals to access directly on the Power Macintosh? Hint: is $16A divisible by 4?

Food For Thought

"Remove these items because they can cause clutter and use a large amount of memory: QuickDraw GX and PowerTalk."
– *from a product installation guide*



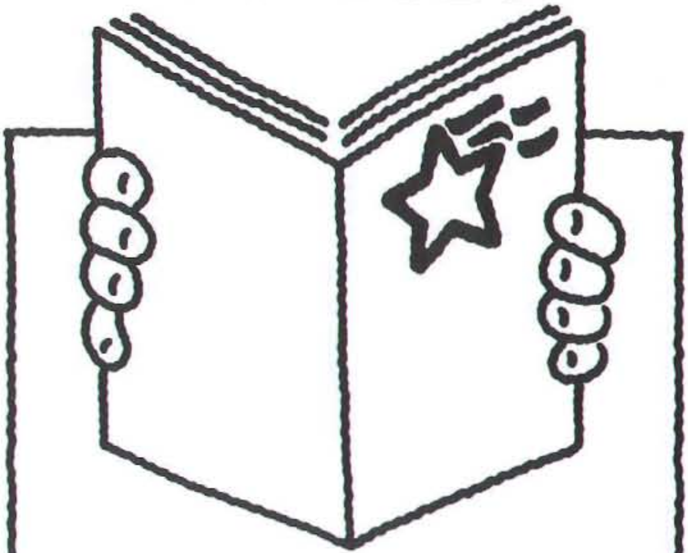
Did You Know?

You can have full code listings every month with a disk subscription. See the Mail Order Store section for rates.

Call, fax, or write to order your disk subscription.

See How to Communicate with Xplain Corporation on page 2 for contact info.

This could be a best seller.



But it's free.

You can't buy the Consumer Information Catalog anywhere. But you can send for it, free!

It's your guide to more than 200 free or low-cost government publications about getting federal benefits, finding jobs, staying healthy, buying a home, educating your children, saving and investing, and more.

Send today for your latest free Consumer Information Catalog. The Catalog is free. The information is priceless.

Send your name and address to:

Consumer Information Center
Department BEST
Pueblo, Colorado 81009

A public service of this publication and the Consumer Information Center of the U.S. General Services Administration

MAIL ORDER STORE

MACTECH MAGAZINE PRODUCTS & ORDER INFORMATION

E-mail, Fax, write, or call us. You may use your VISA, MasterCard or American Express; or you may send check or money order (in US funds only):
MacTech Magazine, P.O. Box 5200, Westlake Village, CA 91359-5200 • Voice: 805/494-9797 • Fax: 805/494-9798

If you are an e-mail user, you can place orders or contact customer service at:

- **AppleLink:** MT.CUSTSVC
- **CompuServe:** 71333,1063
- **Internet:** custservice@xplain.com
- **America Online:** MT CUSTSVC
- **GEnie:** MACTECHMAG
- **eWorld:** MT.CustSvc

SUBSCRIPTIONS

US Magazine: $47 for 12 issues
Canadian: $59 for 12 issues
International: $97 for 12 issues
US Magazine with Source Code Disk: $124 for 12 issues
Canadian Magazine w/Source Code Disk: $136 for 12 issues
International Magazine w/Source Code Disk: $194 for 12 issues

CD-ROM

MacTech CD-ROM, Volumes 1-10: Includes over 1230 articles from all 115 issues (1984-1994) of MacTech Magazine (formerly MacTutor). All article text and source code. Now in THINK Reference format. The CD includes Symantec's THINK™ Reference 2.0, working applications with full documentation, product demos for developers and more. See advertisement, this issue: $49. Upgrades $39. E-mail, call or write for info.

BOOKS

The Best of MacTutor, Volume 1: Sold Out
The Complete MacTutor, Volume 2: Sold Out
The Essential MacTutor, Volume 3: $19.95
The Definitive MacTutor, Volume 4: $24.95
The Best of MacTutor, Volume 5: $34.95
Best of MacTutor Collection, Volumes 3 – 5: $69
Best of MacTutor, Volumes 6, 7, 8 & 9: Not available

DISKS

Source Code Disks: $8 each
Topical Index (1984-1991) on disk: $5

MAGAZINE BACK ISSUES

Volumes 3, 4, 5, 6, 7, 8, 9 and 10: $5 each (subject to availability)

California residents include 8.25% sales tax on all software, disks and books.

Allow up to 2 weeks for standard domestic orders, more time for international orders.

PLEASE NOTE

Source code disks and journals from MacTech Magazine are licensed to the purchaser for private use only and are not to be copied for commercial gain. However, the code contained therein may be included, if properly acknowledged, in commercial products at no additional charge. All prices are subject to change without notice.

10% OFF ALL BOOKS!

3RD PARTY PRODUCTS

MACTECH EXCLUSIVES

MacTech Magazine is your exclusive source for these specific products:

Ad Lib 2.0 The premier MacApp 3.0 compatible ViewEdit replacement. A powerful user-interface editing tool to build views for MacApp 3.0 and 3.1. Ad Lib allows subclassing of all of MacApp's view classes including adorners, behaviors, and drawing environments. String and text style resources are managed automatically. Alternate display methods, such as a view hierarchy window, allow easy examination of complex view structures. Ad Lib includes source code for MacApp extensions that are supported by the editor – buttons can be activated by keystrokes, behaviors can be attached to the application object, and general purpose behaviors can be configured to perform a number of useful functions. Run mode allows the user to try out the views as they will work in an application. Templates can be created to add additional data fields to view classes. Editing palettes provide fast and easy editing of common objects and attributes. Works with ACI's Object Master (version 2.0 and later) to navigate a project's user interface source code. $195

FrameWorks Magazine: $8/back-issue, subject to availability.

FrameWorks Source Code Disk: $10 per back issue, subject to availability.

Five Years of Objects CD-ROM: FrameWorks archives and source code from April 1991 to January 1993, plus selected object-oriented publicly available software and demos. $95

MADACON '93 CD-ROM: The highlights of MADACON '93, including Mike Potel on Pink, Bedrock, MacApp, OODLs, and more. Slides, articles, demos, audio, and QuickTime. $95

MAScript 1.2 adds support for AppleScript to your MacApp 3.0.1 and 3.1 based applications. Make your application scriptable and recordable by building on a tried and tested framework for object model support. MAScript dispatches Apple events to the appropriate objects, creates object specifiers, and makes framework objects like windows and documents scriptable and recordable. Sample application shows you how to begin adding support for scripting and recording. MAScript includes complete source code. Install MAScript by modifying one MacApp source file, then adding another to your project. Future versions of MacApp will incorporate MAScript, so MAScript support you add now will work in the future. $199

The Mjølner BETA System is a software development environment supporting object-oriented programming in the BETA programming language. BETA is uniquely expressive and orthogonal. BETA unifies just about every abstraction mechanism – including class, procedure, function, coroutine, process and exception – into the ultimate abstraction mechanism: the pattern. BETA includes: general block structure, strong typing, whole/part objects. The compiler: binary code generation, automatic garbage collection, separate compilation, interface to C, Pascal, and assembler. The system: persistent objects, basic libraries with containers classes, platform-independent GUI application frameworks on Unix, Mac and Windows NT, metaprogramming system. The tools available on Unix: the hyper structure editor supporting syntax directed editing, browsing, etc., and the source code debugger are currently being ported to the Macintosh system. The Mjølner BETA System for Macintosh requires MPW (basic set) 3.2 or later. Package containing compiler, basic libraries, persistent store, GUI framework, and comprehensive documentation. (Other packages are also available) $295

More Savvy includes all Savvy features plus Apple Event support for all sub-classes of TEventHandler with extensive view support. Apple Event support for text includes text attributes and subrange specification. Recordability supports additional actions, and coercion includes additional types. Additional client and server Apple Events. $450

NEW Version! **Savvy 1.1** OSA support includes attachability, recordability, scriptability, coercion, in addition to script execution, idling and i/o. Apple Event support includes complex object specifiers, synchronous/asynchronous Apple Event handling, and Apple Event transactions for clients and servers. The Core Suite of Apple Event objects is supported including the application, documents, windows, and files. Documentation includes technology overview, cookbook, and sample code. $250 Savvy now supports MPW 3.1, 3.11 and continues to support 3.01, as well as supporting Metrowerks CodeWarrior. **This month only, special offer** – All Savvy versions include free copy of Savvy QuickTime!

Savvy QuickTime Requires Savvy, More Savvy, or Super Savvy. Includes QuickTime, Apple Event, and view template support. Movies come out of the box ready to play, edit, and react to Apple Events. They can be included in any view structure, including templates, and are displayed in the scrap view. Movie controls include volume, play rate, looping mode, display style, and other characteristics. $250

Super Savvy includes all More Savvy features plus compile, edit, and record scripts using built in script editor. View template editors, like Ad Lib, can attach scripts to view objects and modified scripts are saved with the document. Script action behavior allow quick access for executing and editing scripts attached to views. Text to object specifier coercion plus more. $700

MacTech Magazine is your exclusive source for available back issues of SFA's magazine, source code disks and assorted CD's. Call for more info and pricing.

MAIL ORDER STORE

BOOKS

10% OFF ALL BOOKS!

Learn C on the Macintosh by Dave Mark. This self-teaching book/disk package gives you everything you need to begin programming on the Macintosh. Learn to write, edit, compile, and run your first C programs through a series of over 25 projects that build on one another. The book comes with THIN C – a customized version of Symantec's THINK C, the leading programming environment for Macintosh. 464 pages, Book/disk: ~~$34.95~~ **$31.45**

Macintosh C Programming Primer Volume I, Second Edition, Inside the Toolbox Using THINK C by Dave Mark and Cartwright Reed. This new edition of this Macintosh programming bestseller is updated to include recent changes in Macintosh technology, including System 7, new versions of THINK C and ResEdit, and new Macintosh machines. Readers will learn how to use the resources, Macintosh Toolbox and interface to create stand-alone applications. 672 pages, ~~$26.95~~ **$24.25**

Macintosh C Programming Primer Volume II, Mastering the Toolbox Using THINK C by Dave Mark. Volume II picks up where Volume I leaves off, covering more advanced topics such as: Color QuickDraw, THINK Class Library, TextEdit, and the Memory Manager: 528 pgs. ~~$26.95~~ **$24.25**

NEW! **Macintosh OLE2 Programmer's Reference: Working With Objects:** Provides a complete reference to the extensible protocol of Object Linking and Embedding, version 2.01 for Macintosh System 7. Understanding of C/C++ helpful, but not necessary, and comes with a CD. Working With Objects describes the visual and interactice interfaces that support the component objects, provides details of the OLE 2.01 for the Macintosh user Interface, addresses the issues of object class registration, shows how to implement the drag and drop objects from one application to another, covers the interface that exposes the basic embedding functionality, and includes descriptions of API functions. ~~$44.95~~ **$40.45**

C For Dummies,™ Volume 1 by Dan Goodwin. Finally! A hands-on, step-by-step tutorial for learning the essentials of programming in C. ~~$19.95~~ **$17.95**

Macintosh Pascal Programming Primer Volume I, Inside the Toolbox Using THINK Pascal by Dave Mark and Cartwright Reed. This tutorial shows programmers new to the Macintosh how to use the Toolbox, resources, and the Macintosh interface to create stand-alone applications with Symantec's THINK Pascal. 544 pages ~~$26.95~~ **$24.25**

NEW! **Mastering the THINK Class Library** by Richard Parker. Now that Symantec's long-awaited PowerPC native compiler is here, developers are taking another look at THINK. This book provides a thorough examination of Symantec's extensive Class Library and the Visual Architect, a graphic user interface development tool that allows you to produce commercial-quality applications with a minimum of effort. A complete description of the structure and operation of the TCL includes explanations of all code generated by the Visual Architect, any necessary custom code, and the operation of this code. Visual Architect tutorials provide you with a step-by-step approach for simplifying the development of complex Macintosh applications. 496 pages ~~$29.95~~ **$26.96**

C++ For Dummies™ by Stephen R. Davis is the all-in-one reference that gets you comfortable with object-oriented methods and up and running doing C++ programming! ~~$19.95~~ **$17.95**

Learn C++ on the Macintosh by Dave Mark. After a brief refresher course in C, Learn C++ introduces the basic syntax of C++ and object programming. Then you'll learn how to write, edit, and compile your first C++ programs through a series of programming projects that build on one another as new concepts are introduced. Key C++ concepts such as derived classes, operator overloading, and iostream functions are all covered in Dave's easy-to-follow approach. Includes a special version of Symantec C++ for Macintosh. Book/disk package with 3.5" 800K Macintosh disk. 400 pages, ~~$36.95~~ **$33.26**

Programming Primer For The Macintosh® Volume 1 by John Whittle and Judy May. This book provides an introduction to Macintosh programming, using C++ as the example language, and provides realistic, easy to follow, programming examples designed to work with either Symantec® C++ or Metrowerks® CodeWarrior 6™. Also includes one 3.5" disk with source code for the programming examples, along with numerous, useful, public domain utilities to use with each compiler. ~~$37.95~~ **$34.15**

Programming in Symantec C++ for the Macintosh by Judy May and John Whittle. This book will introduce you to object-oriented programming, the C++ language, and of course Symantec C++ for the Macintosh. You don't have to be a programmer, or even know anything about programming to benefit from this book. Programming in Symantec C++ for the Macintosh covers everything from the basics to advanced features of Symantec C++. If you are a Think C or Zortech C++ programmer who wants to learn more about object-oriented programming or what's different about Symantec C++, there are chapters specifically for you. Includes helpful examples of C++ code that illustrate object-oriented programs. ~~$29.95~~ **$26.95**

Symantec C++ Programming for the Macintosh, Second Edition by Neil Rhodes & Julie McKeehan is the perfect introduction to C++ programming – the most popular programming language for the Mac! This updated edition is the easiest way to learn C++ in the Think environment, using many examples and hands-on coding experience. The official Symantec book, developed in cooperation with the Product Development staff. Provides valuable solutions, information, and advice for MPW programmers who are migrating to the Think environment. Disk includes source code from the book and example applications ~~$45.00~~ **$40.50**

Teach Yourself Mac C++ Programming in 21 Days by Namir Clement Shammas is the easy-to-follow 21-day format teaches readers how to program in C++ using the Symantec C++ compiler. It also shows readers how to develop GUI applications using the latest version of the THINK Class Library (TCL). Targets Symantec C++ 7.0, one of the hottest programming languages and development environments. Discusses the basics of programming for the Mac using TCL and the Visual Architect utility. Uses notes, tips, and warnings, as well as Q&A, Quiz, Exercise, and Do/Don't sections to teach users the C++ programming language. ~~$29.99~~ **$26.99**

Writing Localizable Software for the Macintosh by Daniel R. Carter. 469 pages. ~~$26.95~~ **$24.25**

NEW! **Global Interface Design, A Guide to Designing International User Interfaces** by Tony Fernandes, AP Professional. Global Interface Design addresses the issues involved in product development for a global market with a "real world" focus. While covering major areas developers should address during the development cycle, Tony Fernandes provides insight into researching cultural differences. This book examines the differences found all over the world, such as cultural symbolism and taboos, and how they impact user interfaces. ~~$34.95~~ **$32.35**

Taligent's Guide to Designing Programs: Well-Mannered Object-Oriented Design in C++ is the Taligent approach to object-oriented design. The Taligent Operating Environment is the first commercial software system based entirely on object-oriented technology. Taligent's Guide to Designing Programs is a developer's-eye view of this system. It introduces new concepts of programming and empowers developers to create software more productively. Out of their direct experience in developing the system, the authors focus on global issues of object-oriented design and writing C++ programs, and the specific issues of programming in the Taligent Operating Environment. Taligent's Guide to Designing Programs assumes the reader is an experienced C++ programmer, and proceeds from there to fully explore "the Taligent way" of programming. ~~$19.50~~ **$17.55**

Software By Design: Creating User Friendly Software by Penny Bauersfeld (Series Editor: Tony Meadow). This excellent reference provides readers with a thorough how-to for designing software that is easy to learn, comfortable to operate and that inspires user confidence. Written from the perspective of Macintosh, but compatible with all platforms. Stresses user input from initial design, through prototyping, testing and revision. Provides tools for analyzing user needs and test responses. Includes exercises for sharpening user-oriented design skills. ~~$29.95~~ **$26.95**

Macintosh Programming Techniques by Dan Sydow (Series Editor: Tony Meadow). This tutorial and handbook provides a thorough foundation in the special techniques of Macintosh programming for experienced Macintosh programmers as well as those making the transition from DOS, Windows, VAX or UNIX. Emphasizes programming techniques over syntax for better code, regardless of language. Guides the reader through Macintosh memory management, QuickDraw, events and more, using sample program in C++. Disk includes an interactive tutorial, plus reusable C++ code. ~~$34.95~~ **$31.95**

Macworld Ultimate Mac Programming by Dave Mark. Bestselling Mac programming author Dave Mark reveals the secrets of Mac programming and presents important, timesaving techniques. ~~$39.95~~ **$35.95**

Mac Screamer The Ultimate Macintosh® Supercharging Kit by Jan Harrington covers 30 Macintosh models, including the Classics, LCs, PowerBooks, and Quadras and gives software solutions and hardware tips to accelerate Mac performance. It lets readers in on do-it-yourself tips that can save them over 25% on upgrade costs. ~~$35.00~~ **$31.50**

NEW! **A Fragment of Your Imagination** by Joe Zobkiw. Here's some practical help for creating code resources and code fragments for the Macintosh and Power Macintosh. Rather than simply

For complete product info see the MacTech Web™ site at http://www.mactech.com.

gathering and indexing chunks of this vital code, the author provides thorough explanations to teach you more about how the Macintosh system functions as a whole. He also provides hard to find information about techniques used to structure and build fat, safe fat, and accelerated code resources for use on both 680x0 and Power Macintosh. All code is reusable and is provided on the disc, along with Metrowerks Code Warrior Lite. Book/CD-ROM, 528 pages ~~$39.95~~ **$35.96**

Mac Programming for Dummies™ by Dan Parks Sydow takes the intimidation and work out of writing Mac programs. ~~$19.95~~ **$17.95**

Programming for System 7 by Gary Little and Tim Swihart, is a hands-on guide to creating applications for System 7. It describes the new features and functions of the operating system in detail. Topics covered include file operations, cooperative multitasking, Balloon Help, Apple events, and the File Manager. Numerous working C code examples show programmers how to take advantage of each of these features and use them in developing their applications. 384 pages. ~~$26.95~~ **$24.25**

Guide to Macintosh System 7.5 by Don Crabb. Written by an industry expert, this is the first thorough book on the new and improved Macintosh operating system. Readers will learn all the highlights of the new System, including how to work efficiently with applications. Intuitive, task-oriented approach teaches topics the way users think, not the way the machine thinks. Covers all the new features of System 7.5, including PowerTalk, PC Exchange, and MacTCP. Written by a world-renowned Mac expert. ~~$25.00~~ **$22.50**

How To Write Macintosh Software by Scott Knaster is a great source for understanding Macintosh programming techniques. Drawing from his years of experience working with programmers, Scott explains the mysteries and myths of Macintosh programming with wit and humor. The third edition, fully revised and updated, covers System 7 and 32-bit developments, and explores such topics as how and where things are stored in memory; what things in memory can be moved around and when they may be moved; how to debug your applications with MacsBug; how to examine your program's code to learn precisely what's going on when it runs. 448 pgs. ~~$28.95~~ **$26.05**

Danny Goodman's Macintosh® Handbook Featuring System 7 by Danny Goodman with Richard Saul Wurman. This user friendly design includes a unique four-color design and exploded diagrams. It includes over 100 spreads break down and clarify Mac problems and includes insider's tips. ~~$29.95~~ **$26.95**

NEW! **Real World Apple Guide, For The Mac** is the much anticipated help and navigational aid component of the new Apple System 7.5 OS. The book is a practical introduction to Apple Guide for programmers. It explains the design and function of Apple Guide, how to design your own guides using Apple Script. Comes with a disk of sample Apple Guides for Apple Guide-compliant applications. ~~$39.95~~ **$35.95.**

Danny Goodman's Apple Guide Starter Kit by Danny Goodman and Jeremy Joan Hewes. Two highly respected experts offer a different approach for creating your own Apple Guide databases. With Danny's Guide Starter program you can make guides quickly and easily, without having to learn a scripting language, write coded files, or use several different files and programs to produce your database (which is what you'd have to do without the program). The authors provide advice and tips on how to design a good Guide, from planning and creation through testing, revising, and indexing. Book/disk, 320 pages. ~~$34.95~~ **$31.46**

HyperTalk® 2.2: The Book Second Edition by Dan Winkler, Scott Kamins, and Jeanne DeVoto is the most complete, authoritative source on HyperTalk 2.2 programming and troubleshooting. It covers each language element of HyperTalk 2.2 (including the odd quirk or bug). ~~$35.00~~ **$31.50**

The Complete HyperCard® 2.2 Handbook Fourth Edition by Danny Goodman is the biggest-selling Mac book – newly revised and updated for version 2.2. It shows how to build working applications using the latest version of HyperCard and covers text, painting tools, extension commands (XCMDs), scripting in HyperTalk, and more. ~~$35.00~~ **$31.50**

Dan Shafer Presents the Power of Prograph CPX is a hands-on, project-centered approach to learning the most revolutionary object-oriented programming language on the planet. The language Kurt Schmucker likes best. The language that programmers actually report having "fun" programming. This 550-page book takes you step by step through three interrelated projects of increasing complexity. Along the way you'll learn the underlying Prograph language, how to use the power of lists, and the important aspects of the CPX classes and object editors. Includes disk with all code in the book. ~~$49.95~~ **$44.95**

NEW! **Visual Programming With Prograph CPX** by Scott B. Steinman and Kevin G. Carver. This is the first book on Prograph CPX available through the book trade. It covers the only commercially supported visual programming language at a time when many programmers and managers, faced with continuing productivity problems, are searching for better programming environments. Prograph CPX is much more than such GUI-enhanced traditional languages as Visual Basic: It literally allows you to draw your program flow using icons and create a complete application without writing a line of code. This book is an introduction to the language and a guide for advanced users, for both Macintosh and Windows-based machines. Prograph is a fully pictorial, general-purpose, object-oriented language that speeds development with an integrated environment for design, coding, testing and debugging; with its OO framework for sophisticated GUI development; with its support for calls from C, C++, Pascal, and other routines; with its DAL, ORACLE, Sybase, AS/400 client/server DB support; and many other powerful features. Visual Programming With Prograph CPX teaches this new language and environment, offering a careful pedagogic treatment full of useful examples and summaries of needed concepts. ~~$34.00~~ **$30.60**

Graphic Gems V Edited by Alan W. Paeth is the newest volume in The Graphic Gems Series. It is intended to provide the graphics community with a set of practical tools for implementing new ideas and techniques, and to offer working solutions to real programming problems. These tools are written by a wide variety of graphics programmers from industry, academia, and research. The books in this series have become essential, time-saving tools for many programmers. It is the latest collection of graphics tips in The Graphic Gems Series written by the leading programmers in the field. It contains about 50 new gems displaying the most recent and innovative techniques in graphics programming. Also included is new gems in ellipses, splines, Bezier curves, and ray tracing. Includes a disk which contains source code from all five volumes and is available in both IBM and Macintosh versions. CONTENTS: Algebra and Arithmetic. Computational Geometry. Modeling and Transformation. Curves and Surfaces. Ray Tracing and Radiosity. Halftoning and Image Processing. Utilities. ~~$49.95~~ **$44.95**

Danny Goodman's AppleScript Handbook Second Edition by Danny Goodman is a self-contained kit shows the reader how to customize and extend the capabilities of any Macintosh computer – no programming experience needed! This enhanced and expanded edition of The Complete AppleScript Handbook focuses on putting AppleScript to work in all sorts of practical situations. In addition, Danny Goodman's AppleScript Handbook, Second Edition shows you how to apply the same principles to other popular scripting systems, such as UserLand Frontier and QuicKeys. Shows readers how to use scripts to enhance the Macintosh environment, automate many processes, link data between applications, and much more. This book provides a wealth of all-new examples showing how to integrate AppleScript with the Finder, spreadsheets, desktop publishing programs, graphics applications, databases, telecommunications programs, utilities, and HyperCard. The accompanying 3 1/2" disk is jam-packed with over $100 worth of software, including AppleScript 1.1, valuable utilities, and powerful, ready-to-use scripts. ~~$39.00~~ **$35.00**

The Complete AppleScript® Handbook by Danny Goodman is a self-contained kit to customizing and enhancing the Macintosh environment. The disk contains AppleScript 1.1 Runtime, Chang Labs TableServer, and useful, ready-to-run scripts. It also shows the Mac user how to automate many processes – no programming experience necessary. ~~$35.00~~ **$31.50**

The Tao of AppleScript: BMUG's Guide to Macintosh Scripting, Second Edition by Derrick Schneider & Hans Hansen. This updated bestseller is a complete, natural introduction to AppleScript programming essentials. Readers learn how to customize applications, automate tedious tasks, and create programs without having to use a complex programming language. 2 disks contain AppleScript, QuickTime, StuffIt Lite, ResMover, and other helpful utilities. Progressive structure meets the needs of any Mac user, regardless of experience. Professional instructions are mixed with practical examples for easy learning ~~$29.95~~ **$26.95**

Wireless For The Newton Software Development for Mobile Communications by Julie McKeehan and Neil Rhodes is a book that picks up where Programming for the Newton left off, teaching the reader how to develop Newton® software on the Macintosh. The enclosed floppy disk provides a sample application, as well as a fully functional demonstration version of Newton Toolkit™ (NTK™), Apple Computer's complete development environment for the Newton®. Gives hands-on Newton environment training with sample code created specifically for the Newton®. The authors are external faculty at Apple Developer University teaching classes on programming for the Newton®. Programming experience is assumed, although not in any particular language. Enclosed is a floppy disk which contains source code for a Newton application, as well as demonstration NTK™. ~~$34.95~~ **$31.45**

NEW! **Basic For The Newton**, Programming for the Newton Using NS Basic by John Schettino

Want more product info? E-mail us at productinfo@xplain.com

MAIL ORDER STORE

& Liz O'Hara. This book shows owners of Newton devices how to become Newton programmers using BASIC. The authors use a straight-forward "programming by example" approach, which should have you writing your own Newton programs right away. It includes one 3.5" disk containing Demonstration NS BASIC and over fifty example programs from the book. It is Multi-platform in that teaches programming for the Newton using a Macintosh, a Windows-based PC, or on the Newton device itself. ~~$35.95~~ **$32.35**

Programming for the Newton Software Development with NewtonScript by Julie McKeehan and Neil Rhodes. Foreword by Walter R. Smith. Programming for the Newton: Software Development with NewtonScript is an indispensable tool for Newton programmers. Readers will learn how to develop software for the Newton on the Macintosh from people that developed the course on programming the Newton for Apple Computer. The enclosed 3.5" disk contains a sample Newton application from the books, as well as demonstration version of Newton Toolkit (NTK), Apple Computers complete development environment for the Newtons. A Publication of AP Professional May 1994, Paperback, 393 pp. ~~$29.95~~ **$26.95**

Metrowerks CodeWarrior Programming by Dan Parks Sydow. Includes CodeWarrior Lite, and Full Coverage of PowerPlant™. The best information on Metrowerks CodeWarrior 6, giving full coverage to the Gold Edition. Even if you don't already own CodeWarrior 6, you'll still be able to work with the examples in this book, using the CodeWarrior 6 Lite CD that comes with it. ~~$39.95~~ **$35.95**

NEW! **C++ Programming With CodeWarrior** Beginning OOP for the Macintosh and Power Macintosh by Jan L. Harrington from AP Professional. This book shows programming novices object-oriented programming techniques for the Macintosh, Power Macintosh, and Mac OS compatibles, using C++ as the example language and Metrowerks and CodeWarrior as the example compiler. The enclosed CD-ROM contains example code from the book and a full-function Metrowerks CodeWarrior compiler for running these examples. ~~$35.95~~ **$32.35**

Optimizing PowerPC Code: Programming the PowerPC in Assembly Language – To take full advantage of the potential of the PowerPC, Developers need to master the Assembly Language techniques. This book shows how to use the Assembly Language in PowerPC Programs to produce faster more robust software. ~~$39.95~~ **$35.96**

Power Macintosh Programming Starter Kit by Tom Thompson. This is the first tutorial/reference for programmers who want to enter the new world of the PowerPC chips. Users find all the details on the new microprocessors, the new RISC architecture, and how to write native code and emulation operations to create their own software for the Macintosh PowerPC. CD-ROM includes a unique compiler for writing code easily. The all-in-one book that gets programmers the information and tools they need. Programming examples reinforce explanations of code and programming tools ~~$39.00~~ **$35.10**

PowerBook™ The Digital Nomad's Guide by Andrew Gore and Mitch Ratcliffe. ~~$24.00~~ **$21.60**

The ResEdit All Night Diner by David Ciskowski. An idea-filled menu and introduction to the joys of customizing software – and adding personality to the Mac with ResEdit! Shows readers how to customize default icons, the text of menus and dialog boxes, cursors, pointers, and more. Provides specific recipes for doing creative things with ResEdit – plus how to avoid problems. Disk features ResEdit program, plus lots of sample resources ~~$24.95~~ **$22.45**

ResEdit™ Complete, Second Edition by Peter Alley and Carolyn Strange. With ResEdit, Macintosh programmers can customize every aspect of their interface form creating screen backgrounds and icons to customizing menus and dialog boxes. 608 pages. Book/disk package. ~~$34.95~~ **$31.45**

Sad Macs, Bombs, Disasters and What to Do About Them by Ted Landau comes to the rescue with your Macintosh problems. From fractious fonts to the ominous Sad Macintosh icon, this emergency handbook covers the whole range of Macintosh problems: symptoms, causes, and what you can do to solve them. 640 Pages. ~~$24.95~~ **$22.45**

Macintosh® Crash Course by Glenn Brown shows Macintosh power users what to do when things go wrong with their system. Macintosh Crash Course shows readers how to overcome Macintosh system crashes, system lock-ups, and various, frustrating and cryptic error messages they regularly encounter. It includes a CD-ROM with shareware and freeware to help the user diagnose and repair system failures. Includes up-to date coverage through Macintosh System 7.5, Managing memory, Hardware diagnostics, File recovery, PowerBook problems, PowerPC problems, network utilities, hard drive repair utilities, SCSI problems, conflicts and solutions and File synchronization and utilities. ~~$29.95~~ **$26.95**

Multimedia Authoring: Building and Developing Documents by Scott Fisher addresses the concerns that face anyone trying to create multimedia documents. It offers specific advice on when to use different kinds of information architecture, discusses the human-factors concepts that determine how readers use and retain information, and them applies these findings to multimedia documents, covering the high-level issues concerning planners and authors of multimedia documents as well as those involved in evaluating or purchasing multimedia platforms. Includes one 3.5" high-density disk. ~~$34.95~~ **$31.45**

Multimedia Starter Kit for Macintosh by Michael D. Murie. This hands-on book offers the latest and greatest in multimedia for the Mac! Readers learn how to design their own multimedia projects step by step, then try it themselves with the demos, graphics, clips, and sample projects on the CD-ROM! CD-ROM contains QuickTime, sound and graphics clips and utilities, sample projects, and more. How to choose and use a variety of Macintosh multimedia tools and presentation environments. Includes demos of Adobe Illustrator, Premiere, Heizer Software programs, and more ~~$30.00~~ **$27.00**

QuickTime Starter Kit for Macintosh by Robert A. Lettieri & Judith Stern. This is the ultimate package for getting productive and having fun with Macintosh movie-making. Easy steps and valuable software help readers play, make, and edit QuickTime movies. CD-ROM includes QuickTime tools, movie clips, shareware, and demos of Premiere and other programs. Written by members of the respected Berkeley Macintosh User Group. Tips on the best ways to bring live-action video to Mac multimedia ~~$45.00~~ **$40.50**

Adobe Premiere for Macintosh: Classroom in a Book, Second Edition by Adobe Press. This is the updated edition of the official Adobe training workbook! Covering all the essential features of this video editing software, this book/CD-ROM set features twenty tutorial lessons to guide teachers and students through the magic of Macintosh movie-making. CD-ROM includes QuickTime movies, electronic images, and frames needed for the workshop lessons in the book. Detailed yet easy-to-follow steps to creating multimedia presentations, animation, and videos. Contains the latest information on filters, audio, superimposed clips, advanced editing, and more ~~$49.95~~ **$44.95**

3-D Starter Kit for Macintosh by Sean Wagstaff. The complete reference to 3-D graphics on the Macintosh – ideal for beginning to intermediate product designers, illustrators, graphic designers, multimedia developers, animators, and video producers, as well as architects and engineers! Covers more than 50 major Macintosh 3-D imaging software packages – the most comprehensive book available. Lots of idea-packed examples that illustrate how 3-D products work – individually and together. CD-ROM includes sample models, image galleries, backgrounds, and textures, plus 3-D software tryout versions ~~$40.00~~ **$36.00**

Net Chat™ by Michael Wolff & Co. There are an estimated 35 million people online and most of them spend time chatting on the Net. Net Chat is the first comprehensive, discriminating guide devoted to the vast world of online interpersonal communications. Following the successful format of the ever popular Net Guide, Net Chat is designed for easy access, featuring an attractive layout, informative illustrations, and thousands of choices. It is encyclopedic in scope – covers "chat" groups on the Internet, all the major commercial services (including CompuServe, Prodigy, and America Online), and hundreds of bulletin boards. It features a map of salons and meeting places in Cyberspace, where people stop to chat about politics, lifestyle, music, dating, sex, fantasy, health, family, and just about everything imaginable! The entries include one- and two-page spreads that describe the featured topic, and provide easy-access instructions, complete listings, and visual images from the Net. For use with all computer platforms – PCs, Macs, workstations. ~~$19.00~~ **$17.00**

MORE Internet For Dummies™ by John Levine & Margaret Levine Young. The expanded guide full of great Internet tips for all those users who want to know where to go and what to do once connected. ~~$19.95~~ **$17.95**

The Instant Internet Guide by Brent Heslop and David Angell. An Internet jump-start – how to access, use and navigate global networks. The Instant Internet Guide equips readers with the tools needed to travel the electronic world. The book highlights the most important sources of Internet news and information and explains how to access information on remote systems. It outlines how to use essential Internet utilities and programs and includes a primer on UNIX for the Internet. 224 pages ~~$14.95~~ **$13.45**

The Internet by Paul Hoffman gives the straight scoop on the Net with the elegant, entertaining 4-color companion to the PBS special "The Internet Show." Color photos and illustrations explain Internet key features, its history, and trivia. Covers the basics and also a multitude of Internet Information Services and extensive lists of Internet sites. ~~$24.99~~ **$22.50**

The Internet, Deluxe Edition by Paul Hoffman. All

Want more product info? Call us at 805/494-9797.

of the brilliance of The Internet plus the hot NetManage Internet Chameleon software! This bundled Deluxe Edition includes everything you need to surf the Net! Fully automated, NetManage has put together 3 disks filled with Internet tools, service providers, applications, and more – an unbelievable value! $34.99 $31.50

The Internet For Macs For Dummies™ by Charles Seiter. Now novice Mac users have an Internet guide written especially for them – in the easy ...For Dummies™ style they love. ~~$19.99~~ **$17.95**

The Internet For Dummies™ 2nd Edition by John Levine and Carol Baroudi. Surf the net with ease using the up-to-the-minute new edition of the #1 bestselling Internet reference – now with friendly help on connecting to the Internet. ~~$19.99~~ **$17.99**

The Internet For Dummies™ Quick Reference by John Levine. This fact-filled quick reference provides plain English explanations of Internet terms and basics. Cross-referenced to "The Internet For Dummies". ~~$8.95~~ **$8.05**

The Internet For Macs For Dummies™ Starter Kit by Charles Seiter. Access and navigate the Net like a pro the fun and easy way with this all-inclusive Starter Kit for Mac users. Kit includes a special edition of the bestselling The Internet For Macs For Dummies™, plus two disks featuring The Pipeline and TCP/Connect II® Extended software with: E-Mail, UseNet News Reader, FTP, Telnet, Gopher, and more! ~~$29.99~~ **$26.99**

Internet Power Tools by John Ross is designed for intermediate PC users, Internet Power Tools is a complete book/disk package that allows access to the world's largest network – the Internet – with the same ease as one uses Windows. The detailed instructions allow readers to find their way around the Internet fast and covers e-mail, file transfers, remote logon, on-line directories, and more. The accompanying disk is packed with powerful utilities, including Cello, PC Eudora, and Panda – easy-to-use graphical user interfaces (GUI) that obviate the need for the obscure Unix line commands that everyone else must learn. It also provides access to even more software via Random House's Internet FTP site and shows how to connect to the Internet via a network connection or dial-in. ~~$40.00~~ **$36.00**

The 1994 Internet White Pages by Seth Godin & James S. McBride is the one and only complete alphabetical directory of people on the Internet. ~~$29.95~~ **$26.95**

Internet SECRETS by John Levine & Carol Baroudi gives the most from the Net with this performance-oriented book – for Windows, UNIX, DOS, and Mac Internet users who know how to get connected but want to optimize their connections. ~~$39.99~~ **$35.99**

NEW! **Web Head; The Mac Guide to the World Wide Web** by Mary Jane Mara. Published by PeachPit Press. This is a beginning to intermediate book that shows you how to get the most from the Web, using plain talk that beginners will understand, and online veterans will appreciate. Web Head also includes in-depth tours of the three leading Web browsers for the Macintosh: Netscape, MacWeb, and MacMosaic, and explains the basics behind all Web browsers – so you will be ready even when the next hot Web browser hits the scene. It will also tell you how to get connected to the Web, finding your way across webspace, fine-tuning your Mac for top Web performance, and how to turn your Mac into a multimedia Web browser. There is also instruction on how to build your own home page, posting pages on the Web, and avoiding common HTML mistakes. ~~$24.95~~ **$22.45**

NEW! **The Underground Guide to Telecommuting** by Woody Leonhard, Addison Wesley. There's no place like home. Especially when your boss, your kid, and the neighbor's dog are all barking for your attention simultaneously. Working away from a corporate office presents great (often unexpected) challenges and offers even greater rewards. Woody Leonhard takes on the toughest aspects of telecommuting and gives you the straight scoop on how to make it work for you. Whether you're a telecommuter, a telecommuter's boss, or just curious, The Underground Guide to Telecommuting will give you the tools and information you need to turn electricity and a phone line into major productivity. ~~$24.95~~ **$22.45.**

The Elements of E-Mail Style by Brent Heslop and David Angell. Learn the rules of the road in the e-mail age. Concise, easy-to-use format explaining essential e-mail guidelines and rules. It covers style, tone, typography, formatting, politics and etiquette. It also outlines basic rules of composition within the special context of writing e-mail and includes samples and templates for writing specific types of e-mail correspondence. 208 pages. ~~$14.95~~ **$13.45**

E-Mail Essentials by Ed Tittel & Margaret Robbins is a hands-on guide to the basics of e-mail, the ubiquitous networks communication system. The book is suitable for both the casual e-mailer and the networking professional, as it covers everything from the installation of e-mail to the maintenance and management of e-mail hubs and message servers. The books explains the fundamental concepts and technologies of electronic mail, featuring chapters on Lotus applications and CompuServe, as well as information on upgrading, automation, message-based applications, and user training. E-mail Essentials is a step-by-step, jargon-free guide that will enable the e-mail user to get the most out of the communication potentials of networking. 250 pp. ~~$24.95~~ **$22.45**

NEW! **The Computer Privacy Handbook** is a practical guide to e-mail encryption, data protection, and PGP privacy software. With millions of e-mail messages and on-line discussions exchanged daily on the Internet, electronic security has become a key concern. The Computer Privacy Handbook explains practical steps individuals can take to safeguard their electronic security. It also gives a vivid description of how the "Surveillance Age" threatens each person's personal security, a non-technical introduction into data encryption, and the U.S. Government's Clipper Chip surveillance proposal. It also provides an overview of PGP, (Pretty Good Privacy), the world standard for e-mail privacy. ~~$24.95~~ **$22.45.**

America Online For Dummies™ by John Kaufeld. "Driver's Education" for this wildly popular on-line service – covering everything from the main menu to the mail groups for Windows, DOS, and Mac platforms. Includes a coupon for free usage time on AOL for first-time users. ~~$19.99~~ **$17.95**

CompuServe For Dummies™ by Wallace Wang. Find out how to shop, play games, join forums, get the latest news, do research, and more on this popular on-line service. ~~$19.95~~ **$17.95**

eWorld: The Official Guide for Macintosh Users by Cary Lu & John Milligan. Discover the next generation of online services – Apple's eWorld. Certain to be a bestseller, this authorized guide to contains the exclusive Apple software needed to connect to eWorld. It also provides a road map of eWorld and all of its features, and is the first and best book on this exciting new service. Covers every aspect of eWorld, the next generation of online services. Everything users need to get connected and get the most out of eWorld. The only official Apple book and software for accessing eWorld. ~~$29.95~~ **$26.95**

NEW! **PowerPC System Architecture** by MindShare. This book describes the hardware architecture of PowerPC systems, providing a clear, concise explanation of the PowerPC specification, the template upon which all PowerPC processors are designed. The author provides a complete description of the specification for both the 32- and 64-bit implementations. 656 pages ~~$34.95~~ **$31.46**

NEW! **PCI System Architecture, Third Edition** by MindShare. Describing revision 2.1 of the Peripheral Component Interconnect (PCI) bus specification, this book explores PCI's relationship to the rest of the system. It includes an in-depth treatment of PCI to PCI bridges, the PCI BIOS, the 66MHz PCI bus, and more. 592 pages. ~~$34.95~~ **$31.46**

NEW! **Cyberpunk Handbook, The Real Cyberpunk Fakebook** by St. Jude, R.U.Sirius, and Bart Nagel. Published by Random House. This book tells how to tell if *you* or *someone* you know is a Cyberpunk. ~~$9.95~~ **$8.95**

THE APPLE LIBRARY

NEW! **Advanced Color Imaging On the Mac OS** explains how you can augment the color support supplied with Quickdraw, and QuickdrawGX, using the Palette manager to provide the best set of colors on displays with limited color capabilities, soliciting the color choices from users with the color Picker Manager/ Using the ColorSync manager to match colors between screens and input and output devices such as scanners and printers/ learning how the color Manager assists Color QuickDraw in mapping an applications color requests to the actual colors available. ~~$36.95~~ **$33.25.**

NEW! **3D Graphics Programming Using QuickDraw 3D** by Apple Computer, Inc. Now you can incorporate spectacular 3D graphics into your applications. This book/CD-ROM package explores QuickDraw 3D, a graphics extension to the Mac OS for Power Macintoshes. The CD contains the complete QuickDraw 3D system itself and a complete database of the QuickDraw 3D API, allowing you instant access to the hundreds of graphics calls via a fast viewing engine. Book/CD-ROM, 640 pages. ~~$39.95~~ **$35.96**

NEW! **Apple Guide Complete** by Apple Computer, Inc. For those who want the full power of Apple's complete toolset, this book and CD-ROM package from Apple provides everything you need to produce guide files successfully, including Guide Maker, the software you use to build and test guide files. You'll learn about the complete cycle of designing as well as advanced topics such as scripting and coding guide files. Book/CD-ROM, 544 pages. ~~$39.95~~ **$35.96**

Inside AppleTalk by Gursharan S. Sidhu, Richard F. Andrews and Alan B. Oppenheimer. Apple Computer, Inc. 650 pages. ~~$34.95~~ **$31.45**

AppleScript Finder Guide, English Dialect, by Apple

For complete product info see the MacTech Web™ site at http://www.mactech.com.

Computer, Inc. ~~$19.95~~ **$17.95.**

AppleScript Language Guide, by Apple Computer, Inc. ~~$29.95~~ **$26.95.**

AppleScript Scripting Additions Guide, by Apple Computer, Inc. ~~$18.95~~ **$17.05.**

HyperCard Stack Design Guidelines by Apple Computer, Inc. is an essential book for everyone who creates Apple HyperCard stacks, from beginners to commercial developers. It covers the basic principles of design that, when incorporated, make HyperCard stacks effective and usable. Topics include guidelines, navigation, graphic design and screen illustration, text in stacks, music and sound, a sample stack development scenario, collaborative development, and the Stack Design Checklist. 240 pages. ~~$21.95~~ **$19.95**

Macintosh Programmer's Toolbox Assistant CD-ROM – Instant electronic access to Inside Macintosh essentials. Now Macintosh programmers can get quick access to over 4,000 Toolbox calls that are at the heart of Macintosh system software. The definitions of these data structures, resources, constants, and functions are documented in the Inside Macintosh series and are essential information for anyone developing Macintosh software. Macintosh Programmer's Toolbox Assistant is a CD-ROM that harnesses the power of one of the best search and viewing engines in the industry. It allows programmers to access the Toolbox calls quickly from their development environment. With hypertext links allowing programmers to view related topics easily. Macintosh Programmer's Toolbox Assistant is the ultimate electronic reference tool for Macintosh programmers. **$99.95**

Inside Macintosh®: CD-ROM by Apple Computer, Inc. Inside Macintosh is the essential reference for programmers, designers, and engineers for creating applications for the Macintosh family of computers. Inside Macintosh CD-ROM collects more than 25 volumes in electronic form, including: QuickDraw™ GX Library, Macintosh Human Interface Guidelines, PowerPC System Software, Macintosh Toolbox Essentials and More Macintosh Toolbox, QuickTime and QuickTime Components. Now programmers will be able to access over 16,000 pages of the information they need directly from their computers. Hypertext linking and extensive cross referencing across volumes allows programmers to search and explore this library in ways that are unique to the electronic medium. Every Macintosh programmer will regard Inside Macintosh CD-ROM as their most important resource. **$99.95**

Inside Macintosh®: Overview by Apple Computer, Inc. is the first book that people who are unfamiliar with Macintosh programming should read. It gives an overview of Macintosh programming fundamentals and a road map to the New Inside Macintosh library. Inside Macintosh: Overview also covers various programming tools and languages, compatibility guidelines and an overview of considerations for worldwide development. 176 pages. ~~$22.95~~ **$20.65**

Inside Macintosh®: Files by Apple Computer, Inc. describes the parts of the operating system that allow you to manage files. It shows how your application can handle the commands typically found in a File menu. It also provides a reference to the File and Alias Managers, the Disk Initialization and Standard File Packages. 510 pgs. ~~$29.95~~ **$26.95**

Inside Macintosh®: Operating System Utilities by Apple Computer, Inc. describes parts of the Macintosh Operating System that allow you to manage various low-level aspects of the operating system. Everyone who programs the Macintosh should read this book! It will show you in detail how to get information about the operating system, manage operating system queues, handle dates and times, control the settings of the parameter RAM, manipulate the trap dispatch table, and receive and respond to low-level system errors. ~~$26.05~~ **$23.45**

Inside Macintosh®: Processes by Apple Computer, Inc. describes the parts of the Macintosh operating system that allow you to control the execution of processes and interrupt tasks. It shows in detail how you can use the Process Manager to get information about processes loaded in memory. It is also a reference for the Vertical Retrace, Time, Notification, Deferred Task, and Shutdown Managers. 208 pages. ~~$22.95~~ **$20.65**

Inside Macintosh®: Memory by Apple Computer, Inc. describes the parts of the Macintosh operating system that allow you to manage memory. It provides detailed strategies for allocating and releasing memory, avoiding low-memory situations, reference to the Memory Manager, the Virtual Memory Manager, and memory-related utilities. 296 pages. ~~$24.95~~ **$22.45**

Inside Macintosh®: AOCE Application Interfaces by Apple Computer, Inc. shows how your application can take advantage of the system software features provided by PowerTalk system software and the PowerShare collaboration servers. Nearly every Macintosh application program can benefit from the addition of some of these features. This book shows how you can add electronic mail capabilities to your application, write a messaging application or agent, store information in and retrieve information from PowerShare and other AOCE catalogs, add catalog-browsing and find-in-catalog capabilities to your application, write templates that extend the Finder's ability to display information in PowerShare and other AOCE catalogs, add digital signatures to files or to any portion of a document, and establish an authenticated messaging connection. ~~$40.45~~ **$36.40**

Inside Macintosh®: AOCE Service Access Modules by Apple Computer, Inc. describes how to write a software module that gives users and PowerTalk-enabled applications access to a new or existing mail and messaging service or catalog service. This book shows how to write a catalog service access module (CSAM), a messaging service access module (MSAM), and AOCE templates that allow a user to set up a CSAM or MSAM and add addresses to mail and messages. ~~$26.95~~ **$24.25**

Inside Macintosh®: Devices by Apple Computer, Inc. describes how to write software that interacts with built-in and peripheral hardware devices. With this book, you'll learn how to write and install your own device drivers, desk accessories, and Chooser extensions; communicate with device drivers using the Device Manager; access expansion cards using the Slot Manager; control SCSI devices using SCSI Manager 4.3 or the original SCSI Manager; communicate directly with Apple Desktop Bus devices; interact with the Power Manager in battery-powered Macintosh computers; and communicate with serial devices using the Serial Driver. ~~$29.95~~ **$26.95**

Inside Macintosh®: Macintosh Toolbox Essentials by Apple Computer, Inc. covers the heart of the Macintosh. The toolbox enables programmers to create applications consistent with the Macintosh "look and feel". This book describes Toolbox routines and shows how to implement essential user interface elements, such as menus, windows, scroll bars, icons and dialog boxes. 880 pages. ~~$34.95~~ **$31.45**

Inside Macintosh®: More Macintosh Toolbox by Apple Computer, Inc. covers other Macintosh features such as how to support copy and paste, provide Balloon Help, play and record sound and create control panels are covered in this volume. The managers discussed include Help, List, Resource, Scrap and Sound. ~~$34.65~~ **$31.45**

Inside Macintosh®: Networking by Apple Computer, Inc. describes how to write software that uses AppleTalk networking protocols. It describes the components and organization of AppleTalk and how to select an AppleTalk protocol. It provides the complete application interfaces to all AppleTalk protocols, including ATP (AppleTalk Transaction Protocol), DDP (Datagram Delivery Protocol), and ADSP (AppleTalk Data Stream Protocol), among others. ~~$29.95~~ **$26.95**

Inside Macintosh®: Interapplication Communication by Apple Computer, Inc. shows how applications can work together. How your application can share data, request information or services, allow the user to automate tasks, communicate with remote databases. ~~$34.95~~ **$31.45**

Inside Macintosh®: PowerPC Numerics by Apple Computer, Inc. describes the floating-point numerics environment provided with the first release of PowerPC processor-based Macintosh computers. The numerics environment conforms to the IEEE standard 754 for binary floating-point arithmetic. This book provides a description of that standard and shows how RISC Numerics compiles with it. This book also shows programmers how to create floating-point values and how to perform operations on floating-point values in high-level languages such as C and in PowerPC assembly language. ~~$28.95~~ **$26.00**

Inside Macintosh®: PowerPC System Software by Apple Computer, Inc. describes the new process execution environment and system software services provided with the first version of the system software for Macintosh on PowerPC computers. It contains information you need to know to write applications and other software that can run on the PowerPC. PowerPC System Software shows in detail how to make your software compatible with the new run-time environment provided on PowerPC-based Macintosh computers. It also provides a complete technical reference for the Mixed Mode Manager, the Code Fragment Manager, and the Exception Manager. ~~$24.95~~ **$22.45**

Inside Macintosh®: Sound by Apple Computer, Inc. describes the parts of the Macintosh system software that allow you to manage sounds. It contains information that you need to know to write applications and other software that can record and play back sounds, compress and expand audio data, convert text to speech, and perform other similar operations. ~~$26.95~~ **$24.25**

Inside Macintosh®: Text by Apple Computer, Inc. describes how to perform text handling, from simple character display to multi-language processing. The Font, Script, Text Services, and Dictionary Managers are all covered, in addition to QuickDraw Text, TextEdit, and International and Keyboard Resources. ~~$39.95~~ **$35.95**

Inside Macintosh®: Imaging by Apple Computer, Inc. covers QuickDraw and Color QuickDraw. The book includes general discussions of drawing and working with color. It describes the structures that hold images and image information, and the routines that manipulate them. It also covers the Palette, Color, and Printing Managers,

Want more product info? Call us at 805/494-9797.

and the Color Picker, Color Matching, and Picture Utilities. ~~$26.95~~ **$24.25**

Inside Macintosh®: QuickDraw™ GX Graphics by Apple Computer, Inc. shows readers how to create and manipulate the fundamental geometric shapes of QuickDraw GX to generate a vast range of graphic entities. It also demonstrates how to work with bitmaps and pictures, and specialized QuickDraw GX graphic shapes. ~~$26.95~~ **$24.25**

Inside Macintosh®: QuickDraw™ GX Objects by Apple Computer, Inc. introduces QuickDraw GX and its object structure, and shows programmers how to manipulate objects in all types of programs. ~~$26.95~~ **$24.25**

Inside Macintosh®: QuickDraw™ GX Environment and Utilities – A companion to QuickDraw™ GX Objects, this book contains programming information useful to any developer writing QuickDraw GX applications. It describes QuickDraw GX memory management, error handling, debugging, and mathematical functions, as well as conversion from QuickDraw to QuickDraw GX. ~~$29.95~~ **$26.95**

Inside Macintosh®: QuickDraw™ GX Library by Apple Computer, Inc. is the powerful new graphics architecture for the Macintosh. Far more than just a revision of QuickDraw, QuickDraw GX is a unified approach to graphics and typography that gives programmers unprecedented flexibility and power in drawing and printing all kinds of shapes, images, and text.

Inside Macintosh®: QuickDraw™ GX Printing This book is essential for any developer whose QuickDraw™ GX application supports printing. It shows how to support the new printing features of QuickDraw GX, including desktop printers and expandable printing dialog boxes. QuickDraw GX Printing also shows how to use printing-related objects to add custom panels to printing dialog boxes and to create custom page formats. ~~$26.95~~ **$24.25**

Inside Macintosh®: QuickDraw™ GX Printing Extensions and Drivers – Any developer who wants to create extensions to the application printing capabilities of QuickDraw™ GX, or who needs to write a printing device driver that works with QuickDraw GX needs this book. QuickDraw GX Printing Extensions and Drivers describes how to create printing extensions and printer drivers, and provides a complete reference to the messages, functions, and resources that they use. ~~$29.95~~ **$26.95**

Inside Macintosh®: QuickDraw™ GX Programmer's Overview – This book provides an introduction to QuickDraw™ GX, providing an overview of the QuickDraw GX environment from a developer's perspective. It introduces the QuickDraw™ GX programming and runtime environments, the relationship between QuickDraw GX and the rest of the Macintosh® systems software and the relationship between QuickDraw GX and Macintosh applications. The key elements of QuickDraw GX programming, data structures, object types, and functions used most frequently by QuickDraw GX developers are also covered. After a general introduction, this book provides readers with a series of practical examples demonstrating how to approach programming with QuickDraw GX. ~~$24.95~~ **$22.45**

Inside Macintosh®: QuickDraw™ GX Typography – This book is essential for any developer who uses QuickDraw™ GX to manipulate text. It shows how to use QuickDraw GX objects to handle all kinds of text – from plain, unstyled text to complex, mixed-direction and multi-language text with sophisticated stylistic and typographic variations. QuickDraw GX Typography shows how to create and manipulate the three different types of text shapes supported by QuickDraw GX including text shapes, glyph shapes, and layout shapes. ~~$29.95~~ **$26.95**

Inside Macintosh®: QuickTime by Apple Computer, Inc. is for anyone who wants to create applications that use QuickTime, the system software that allows the integration of video, animation, and sounds into applications. This book describes all of the QuickTime Toolbox utilities. In addition, it provides the information you need to compress and decompress images and image sequences. ~~$29.95~~ **$26.95**

Inside Macintosh®: QuickTime Components by Apple Computer, Inc.covers how to use and develop QuickTime components such as image compressors, movie controllers, sequence grabbers, and video digitizers. ~~$34.95~~ **$31.45**

Inside Macintosh®: X-Ref by Apple Computer, Inc. is a fast access to all the information in Inside Macintosh. Inside Macintosh X Ref; Provides programmers with a quick and easy way to find the exact information they need in this definitive suite of books, (all 26 volumes). It is indexed by topic, volume, chapter, and accompanying page number. ~~$19.95~~ **$17.95.**

Inside the Macintosh Communications ToolBox by Apple Computer. This book is the definitive reference to the Macintosh Communications Toolbox, an integral part of the System 7 Macintosh Toolbox that enables developers to create communications applications or add communications features to other applications.This book describes all of the routines that provide programmers with standard access to important communications services and in addition enables programmers to extend the reach of the Macintosh into non-Apple environments. ~~$24.95~~ **$22.45.**

EDITORS

BBEdit 3.1 from Bare Bones Software is now better than ever. In addition to being Accelerated for Power Macintosh, this powerful, intuitive text editor offers integrated support for THINK C 7.0, Metrowerks CodeWarrior 6, THINK Reference 2.0 and MPW ToolServer. Version 3.1 adds even more capability, including "soft" wrapping of text on screen and numerous refinements and improvements to the user interface. BBEdit's many features include: Integrated PopupFuncs™ technology for speedy navigation of source code files (C, C++, Pascal, Rez, 68K Assembler, and Fortran), unique 'Find Differences' command (BBEdit can find differences between projects and folders as well as files), support for Macintosh Drag and Drop for editing and other common tasks, PowerTalk support for reading, sending and composition of PowerTalk mail, scripting via any OSA compatible scripting language including AppleScript and Frontier 3.0, and fast search and replace with optional "grep" matching and multi-file searching. BBEdit's robust feature set and proven performance and reliability make it the editor of choice for professionals and hobbyists alike. $99

New Low Price!

CMaster 2.0 by Jersey Scientific installs into THINK C 5 / 6 / 7 and Symantec C++ for Macintosh, and enhances the editor. Use its function popup to select a function and CMaster takes you right to it. Other features include multiple clipboards and markers, a Function Prototyper, and a GoBack Menu which can take you back to previous editing contexts. Almost all features bindable to the keyboard, along over a hundred keyboard-only features like "Add New Automatic Variable." Glossaries, AppleScript and ToolServer support, Macros, and External Tools you create too! $129.95

Want more product info? E-mail us at productinfo@xplain.com

QUED/M 2.7 by Nisus Software, is a programmer's text editor which has defined the industry standard for speed and efficiency. With integrated support for Symantec C/C++, Metrowerks CodeWarrior 6, and MPW, QUED/M offers unrivaled usefulness for the Macintosh developer. In addition to supporting all the major development environments on the Macintosh, QUED/M offers dozens of powerful editing features, including unlimited undo and redo, UNIX style GREP searching, macro language, scripting, text folding, text sorting, file comparison and merging, Toolbox lookup, ten editable/appendable clipboards, line numbering, markers, displaying text as ASCII codes, vertical and horizontal screen splitting, plus much more. $149

DEVELOPMENT ENVIRONMENTS AND LANGUAGES

CodeWarrior™ 7 CD by Metrowerks comes in two versions – Bronze and Gold. These CDs contain the CodeWarrior 6 development environment including C++, C and Pascal compilers; high-speed linkers; native-mode interactive debuggers; and a powerful new application framework called PowerPlant for rapid Macintosh development in C++. Bronze generates 680x0 code. Gold generates both 680x0 and PowerPC code. All versions are a 3 CD subscription over a 1-year period. Bronze: $149, Gold: $399. **Bronze comes with a 6-month MacTech subscription. Gold comes with a 1-year subscription. Both at no additional charge!**

Symantec C++ for Macintosh is an object oriented development environment designed for professional Macintosh programmers. Symantec C++ features powerful object-oriented development tools within a completely integrated environment. The C++ compiler, incremental linker, THINK Class Library, code editor with coloring, integrated browser, debugger with stack crawl, and automatic project management give Symantec C++ fast turnaround times. This product compiles for both PowerPC and 68K machines. Symantec C++ supports multiple editors and translators, so you can use your favorite tools and resource editors as well as scripts you've written within the environment. And with ToolServer, you'll be able to customize menus and attach scripts based on Apple events, AppleScript, and MPW Tools. The built-in SourceServer provides a source code control system, allowing teams of programmers to solve tough problems faster. With SourceServer, you'll always know you're working on the latest version. And you'll have old versions at your fingertips when code "breaks" and you need to look back at modifications. $369

THINK C by Symantec Corporation. THINK C is easy to use and highly visual, making it the No. 1 selling Macintosh programming environment. Enhancements make this product faster and more versatile than ever, improving your productivity with more powerful project management, a full set of tools, and script support for major script-based languages. With the THINK environment, you spend less time on routine programming tasks due to an extremely fast compiler and incremental linker. In addition, the automatic project manager saves you time by tracking changes to your files and recompiling only those that have changes. All the tools you need – a multi-window editor, compiler, linker, debugger, browser, and resource editor – are completely integrated for speed and ease of use. One of the most valuable of these tools is the THINK Class Library, a set of program building blocks that gives you a head start in writing object-oriented applications. And with the new open architecture, you can use your favorite tools, resource editors, and scripts within the environment. THINK C is the logical next step for programmers who have worked in HyperCard or other script-based development environments. The environment supports AppleScript, Apple events, and Frontier, so you can link and automate complex, multi-project operations. Product Contents: Four Macintosh disks, an 832-page user manual, and a 568-page THINK Class Library Guide. $219

THINK Pascal v. 4.0 by Symantec Corporation. Professionals and students will welcome this version of THINK Pascal. It is fully integrated for rapid turnaround time and lets you take advantage of System 7 capabilities. Features include support for large projects, enhanced THINK Class Library, System 7 compatibility, superior code generation, and smart linking. Product Contents: Four Macintosh disks, a 562-page user manual, and a 498-page object-oriented programming manual. $169

LS Object Pascal CD includes the world's first Object Pascal compiler for Power Macintosh. 100% compatible with Apple's MPW Pascal, LS Object Pascal combines the best of Apple's native development tools with innovative new technology developed at Language Systems. Compiler options specify 68K or native PowerPC code generation. Included on the CD are: LS Object Pascal compiler, Universal Pascal Toolbox interfaces, fully loaded MPW 3.3.1, 68K and PowerPC source debuggers, PowerPC assembler, online documentation, Macintosh Tech Notes, and a special version of AppMaker by Bowers Development that generates native Pascal source code. The beta release includes upgrades to v1.0 when it becomes available. $399

LPA MacProlog comprises a Edinburgh syntax Prolog compiler system set in an attractive multi-window development environment with an integrated program editor, graphical call-graph facilities and an interactive source-level debugger. LPA MacProlog features high-level access to the Macintosh ToolBox for using graphics, dialogs, windows, icons, resources in a simple and versatile way LPA MacProlog also includes interfaces to C and Pascal code resources. The MacProlog Run-time Generator enables the production of double-clickable distributable applications. The compact run-time system supports first argument indexing, tail-recursion and last-call optimization. Optional add-ons tools include flex, Prolog++, MacDBI for Oracle and the MacProlog Dialog Editor. Programmer Edition $745 Developer Edition (which includes the run-time generator and distribution license) $1500

Q·K·S

SmalltalkAgents™, a superset of the Smalltalk language, is fully integrated with Macintosh, incor-porating design features specifically for the RISC and Macintosh System 7 architecture. SmalltalkAgents is a true object oriented workbench that includes an incremental and extensible compiler, an array of design and cross reference tools, preemptive interrupt driven threads and events, an extensive class library including classes for general programming, classes for the Macintosh user interface and classes for the Macintosh operating system. Integration of components in enterprise systems is simplified with the network, telecommunication, and inter-application communication libraries. The SmalltalkAgents' extensive class library and add-on components make it especially well suited as a development workbench for custom applications in business, education, science, engineering, and academic research. $695

SOFTWARE ENGINEERING

MacDesigner/Expert by Excel Software supports software engineering methods with the capabilities of MacDesigner plus multi-task design. An integrated requirement database provides traceability from requirement statements to design diagrams, code or test procedures. This tool is well suited to design or maintenance of real-time, multi-tasking software projects. Demo $79, Product $1595. **MacDesigner** Demo $79, Product $995.

MacA&D by Excel Software combines the capabilities of MacAnalyst/Expert and MacDesigner/Expert into a single application. It supports structured analysis and design, object-oriented analysis and design, real-time extensions, task design, data modeling, screen prototyping, code editing and browsing, reengineering, requirement traceability, and a global data dictionary. Demo $149, Product $2995

MacAnalyst/Expert, Demo $79, Product $1595 **MacAnalyst** Demo $79, Product $995 **MacDesigner/Expert,** Demo $79, Product $1595. **MacDesigner** Demo $79, Product $995. By Excel Software. Available. Call for more information about these products.

Voodoo is a version control tool for the simple and clear management of projects in which files are created in numerous versions (variants and revisions). Voodoo allows both variant and revision control, and it manages not only variants and revisions of single files, but of a whole software project (multi files, multi users, multi variants, access rights, ...). The tool offers a neat graphical user interface and is not only suitable for mere source code control but can handle all different kinds of files with amazing compression rates: typical size of delta between arbitrary files 5% (in words: five per cent) !!!! no matter whether the files are plain text or any other documents – e.g., MSWord, 4D, Canvas, FileMaker ...

Please note special prices for multiple copies: single license $190; 2 pack $300; 5 pack $665; 10 pack $1140; 20 pack $2000. Add'l pricing available on request.

PROFILERS / DEBUGGERS

LJ Profiler by Lars Jordebo Datakonsult supports profiling of C++ 68K and PowerPC applications compiled with CodeWarrior 6, CFront or Symantec C++. Based on active profiling, i.e. profiling code called at function enter and exit, the browser application lets you follow call chain timings in hierarchical views or separate windows. Collect, organize, compare and save profiling data from different versions of your application into a project. Scriptable and recordable with full access to most internal data structures. Optional remote profiling and tracking of segment and stack usage. Full source code to what you link into your application. $295.

Last Resort Programmer's Edition records every keystroke, command key and mouse event (in local coordinates) to a file on your hard disk. This is especially useful for program testing & debugging, and for technical support and help desks. If something goes wrong (because of a power failure, system crash, forgetting to save or deleting lines) and you lose a word, phrase, or document you can look in the Last Resort keystroke file and recover what you typed. Last Resort is also useful for technical support personnel, when they have to ask "What was the last thing you did before...?" $74.95

The Memory Mine™ by Adianta is a stand alone debugging tool for Macintosh and native PowerPC. Programmers can monitor heaps, identify problems such as memory leaks, and stress test applications. Active status of memory in a heap is sampled on the fly: allocation in non-relocatable (Ptr), relocatable (Handle) and free space is shown, as are heap corruption, fragmentation, and more... Allocate, Purge, Compact, and Zap memory let users stress test all or part of a program. Source code is not needed to view heaps. It works on Macintoshes with 68020 or later and System 7.0 or later. $99

QC™ by Onyx Technology, is a system extension that stress tests code during runtime for common and not-so-common errors. Tests include heap checks, purges, scrambles, handle/pointer validation, dispose/release checks, write to zero, de-reference zero as well as other tests like free memory invalidation and block bounds checking. QC is extremely user friendly for the non-technical tester yet offers an API for programmers who want precise control over testing. QC is Also available in Japanese. $99.95

Spyer by InCider is a simple operated tool that records all actions (including mouse movement) you perform on a Macintosh computer and then replays them at your preferred speed. The recorded data can be saved in files for future use. Spyer works as a background process with any Macintosh application and is triggered by user defined Hot Keys. Spyer enables the "Continuous Redo" utility and is especially useful for software testing and demonstration. $39

INSTALLER TOOLS

InstallerPack™ by StepUp Software is a package of several Installer "atoms" that let developers incorporate graphics, sounds, file compression and custom folder icons into installation scripts. Compression formats supported are Compact Pro & Diamond. Each atom also available separately. Compression requires additional licensing. $219

ScriptGen Pro™ by StepUp Software is an Installer script generator which requires no programming or knowledge of Rez. Supports StepUp's InstallerPack, StuffIt compression, custom packages, splash screens, network installs, Rez code output, importing resources, and AppleEvent link w/MPW: $169

LIBRARIES

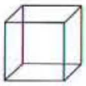
3D Game Machine v1.2 by Virtually Unlimited

Want more product info? Call us at 805/494-9797.

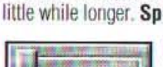
is a C library for creating lightning-fast 3D arcade games and interactive multimedia applications. 3DGM has a simple easy-to-use interface and features very fast rendering (15 frames per second on a 14" monitor completely texture-mapped, with a PowerMac 6100/60), full "virtual" 3D worlds with six degrees of freedom, free-form texture mapping, shading, material and light properties, convex/cave polygons with unlimited vertices, unlimited light sources, dynamic hidden surface removal, special graphic modes for fast full-screen animation, collision detection, explosion simulation, 3D data importing. Runs on all Macs! Works with CodeWarrior. $299 + license.

Animation Class Library version 2.0 (ACL2.0) is an advanced object-oriented multimedia framework, allowing fast development of high-quality interactive applications. Main features of ACL2.0 are: Powerful animation engine which supports structured sprites, collision detection at pixel precision, sprites sorting, powerful blitter and vector objects. Scrolling of background picture in circular buffer and tile-mapscrolling. Application framework for building standard and 3D controls, panes,menus, full screen displays, windows, etc. Quicktime and multi-channel sound support. >800 functions and >100'000 lines. Complete C++ source code for CodeWarrior and Symantec C++, examples, documentation and technical support. $250

dtF is a true relational database system for Apple Macintosh computers. dtF provides a powerful choice for developers who want to create database centered applications with no performance trade-offs. dtF features SQL, full transaction control, error recovery, single user, client server architecture and multi-platform support including DOS, Windows, OS/2 and UNIX. The C/C++ API is identical and fully portable cross all supported platforms. Third-party vendors supporting dtF will be able to offer a variety of advanced features and benefits to their customers royalty free. Tools are included for importing, exporting, creating and managing databases and users. Supported development environments include: Symantec, MPW, Metrowerks and more. Mac/SDK $695



MacWireFrame by Amplified Intelligence. Create your own virtual reality application with MacWireFrame, a virtual reality application frame work. Includes a complete library of object oriented graphics routines, its own easy to understand application frame work (similar to MacApp or TCL but a lot easier to understand), plus an example application program that lets you start solid modeling right away. Comes complete with fully documented source code. All new purchases will be guaranteed a $49.99 upgrade to the soon to be released, scriptable, MacWireFrame 5.0. Due to the overwhelming response the special price offer has been extended for a little while longer. **Special Offer: ~~$299.00~~ $75!!!!**



PictureCDEF 1.3 by Paradigm Software is a professional-level CDEF for creating custom graphical buttons (8-64 pixels). PictureCDEF is used in products by Adobe, ProVue, STF Technologies and others. It is multi-monitor and bit-depth sensitive. The button graphic (cicn, ResEdit) can be changed at runtime and even animated with a call-back routine. Create distinct buttons in seven variations: MultiState, PushButton, FlexiButton, ToggleButton, ChkButton, PushPictButton and TogglePictButton. Position the optional button title at left, bottom or right, or follow the system text direction for international support. Manual, sample code and MacApp 3.0 support included. Full source code: $95.00 Object code: $45.00.

Q3S/3dPane/SmartPane source code bundle by Vivistar Consulting. Q3S: source code bundle from ViviStar Consulting. Full featured 3d graphics. Points; lines; polygons; polyhedra; Gouraud shading; z-buffering; culling; depth cueing; parallel, perspective, and stereoscopic projections; performance enhancing "OnlyQD" and "Wireframe" modes; full clipping; pipeline access; animation and model interaction support; and a "triad mouse" to map 2d mouse movement to 3d. **3dPane** provides integration with the TCL and provides a view orientation controller. **SmartPane** provides TCL offscreen image buffering, flicker free animation, and QuickTime movie recording. SmartPane functions in 3d or 2d scenarios. All work with C++ compilers or ThinkC 6 and compile to PowerPC or 68K target machines. $192

Spellswell 7 1.0.4 is an award-winning, comprehensive, practical spelling checker that works in batch mode or within applications that incorporate the Apple Events Word Services protocol (e.g., Eudora, WordPerfect, Communicate!, and Fair Witness). Spellswell 7 checks for spelling errors as well as common typos like capitalization errors, spaces before punctuation, double double word errors, abbreviation errors, mixed case errors, extra spaces between words, a/an before vowel/consonant, etc... MacTech orders include developer kit with Writeswell Jr., a sample Apple Events Word Services word-processor and its source code. $74.95

StoneTable Extra: Additional functions for StoneTable. Drag selected cells within table or to other tables; optionally add rows as part of drag; popup menus or check boxes in cells; variable width grid lines; move/drag/resize table in window; clipboard operations on multiple cells. Requires StoneTable. (all prices per developer) $50 first compiler, additional compilers $25.

StoneTable: A library replacing all functions found in list manager plus: variable size columns/rows; different font, size, style, forecolor, backcolor per cell; sort, resize, move, copy, hide columns/rows; edit cells/titles in place; titles for columns/rows; multiple lines per cell; grid line pattern/color; greater than 32k data per table; up to 32k text per cell; support for balloon help and binary cell data. Versions for Think C, Think Pascal, MPW C, MPW Pascal, CodeWarrior 6 C. (all prices per developer) $150 first compiler, additional compilers $50.

StoneTable and StoneTableExtra for PowerPC: Same functionality as 68K libraries. Versions for MPW C and CodeWarrior 6 C. Must have 68K libraries. (all prices per developer) StoneTable $100, StoneTableExtra $25.

NEW! **B-Tree HELPER™ 2.2** is an inexpensive database engine for Macintosh programmers in C source code. B-Tree HELPER gets space in a file in contiguous fixed length blocks. It expands the file as necessary and contracts files when possible. B-Tree HELPER inserts and deletes keys in one or more B-Trees. It finds keys equal to, less than, or greater than a given value in a few hundredths of a second. It finds lists of records whose keys are equal to, less than, or greater than a given value or are in a range of values. $150

SCRIPTING / SYSTEM ADMINISTRATION

DataScript DataScript is probably he quickest, easiest and most cost-effective way to make your integrated AppleScript solutions database aware-today. **Quickest:** It takes just six lines of AppleScript to make new or existing scripted solutions database aware, and fetch data from RDBM's such as Oracle, Sybase, DB2, or Informix. **Easiest:** DataScript's scripting terminology is easy to learn, easy to use, and easy to remember. "Inside DataScript" contains lots of easy to follow scripts to reuse in your own solution. **Cost Effective:** Because DataScript is so easy to learn, and use you'll become productive very quickly, and once you're ready to ship, you'll find our licensing schemes very attractive. $249.00

Want more product info? E-mail us at productinfo@xplain.com

CLImate by Orchard Software is a command line interface that lets you communicate with your Macintosh using English commands to create, delete, rename, and move files and folders. It can start applications, format disks, restart your computer, etc. CLImate supplements the Finder. It includes a BASIC interpreter that lets you script your Macintosh without AppleScript. The interpreter includes advanced programming constructs: repeat loops, if/then/else conditionals, subroutine calls, etc... CLImate implements wildcard characters, enabling you to work on groups of files. Use CLImate instead of MPW to manage your projects. CLImate is an application occupying 70K disk space. It comes bundled with sample programs and full documentation. $59.95

Cron Manager by Orchard Software implements the UNIX Cron facility. It can open any Macintosh file on a given date and time. By creating an alias, renaming it to the date and time to open, and moving it into the special Cron Events Folder, Cron Manager will open it. Cron Manager is a control panel that creates the special Cron Events Folder inside your System Folder. It is completely transparent to the user. It works like the Startup Items folder, only smarter. It works with any Macintosh file: if you can double-click to start it, Cron Manager can open it. $26.95. Cron Manager bundled with CLImate, $59.95.

FaceSpan™ v2 is an extensible Rapid Application Designer (RAD) that makes building applications quick and easy. It combines an interactive, visual interface design environment with the object-oriented power of AppleScript or any OSA language. Best of all, FaceSpan allows you to integrate the capability of scriptable programs into your custom application. Your FaceSpan applications can include any number of windows, dialogs, palettes, and menus. In them, you can display scrolling lists, popup menus, scrolling text, movies, multi-column tables, pictures, icons, buttons, and others. While no scripting is needed for standard behaviors, every item may have its own script. You can even program custom objects using Pascal or C. Try the perfect choice for MIS professionals, power users, consultants, and programmers Ø FaceSpan! Includes a royalty-free distribution license, for unlimited runtime users, of your FaceSpan-based applications. Also included is a FREE UPGRADE to the next version for registered users. $199

Rosanne™ Rosanne is a collection of utilities which offer the user complete control over raw data. Users can sort files, extract selected records, summarize frequency counts, create sample files, perform matching on multiple files, and reformat data to new specifications, all on the desktop, and even on files of a million records or more. The Rosanne Utilities also support AppleScript™, enabling the user to link several actions together to complete an entire process. The Rosanne Utilities are recordable; users may perform a series of actions, and using an AppleScript editor such as Scripter™, see their actions translated directly into AppleScript commands. All of the utilities support multi-tasking and background processing. The Rosanne Utilities will assist you in picking your specifications, determining record length, creating output files and managing the storage of data. **Rosanne Utilities: Copy** – duplicates an input file. **Format** – creates an altered version of an input file, containing either subsets of the fields on the input file, or new fields. **Select** – creates a subset of the records on an input file based on some selection criteria. **The Recode** option allows the user to group data, or correct coding entries. **Sort** – orders an input file by a particular field or set of fields. **Match** – joins together two input files based on common values occurring in corresponding fields or sets of fields. **Aggregate** – creates an output file with summary levels. $595

ScriptBase™ The Scripting Database is a database for storing persistent objects to be made available for access to AppleScript, Apple's system-level user scripting language for controlling applications on Macintosh® computers. Once installed, the database becomes part of the AppleScript system, adding a host of commands to the basic AppleScript vocabulary. Retrieving the objects is simple using AppleScript's natural-language syntax and structure. Objects stored and retrieved in ScriptBase can be accessible any time from any script on the user's computer. These objects can be of any type, including numbers, character strings, lists, records, scripts, and references to disks, files, folders, as well as abstract raw data, to name just a few. ScriptBase can be used to maintain system-wide settings, such as sets of preferences, paths to frequently-used files or folders. Complex installations can be made easier by organizing data and scripts within the database's structure. $79

Script Debugger by Late Night Software Ltd. is a powerful and flexible AppleScript authoring tool. Script Debugger makes it simple for novice and experienced script writers to get the most from AppleScript. The program's advanced debugging environment offers single-step script execution with breakpoints. The Script Debugger dictionary browser features a graphical view of objects provided by scriptable applications. With the program, you also receive the Late Night Software Scripting Additions, a collection of more than 70 new AppleScript commands, and Scheduler, a utility that allows you to launch scripts at pre-determined times. $129

Scripter® The Authoring and Development Environment for AppleScript™. Scripter, the Script Construction Set, is the foremost, comprehensive tool for creating and debugging AppleScript scripts. Scripter is a shortcut to Applescript's full capabilities, is both powerful and easy to use, and appeals equally to novices and experts. Scripter offers the largest collection of tools to answer the needs of every AppleScript user, containing over 35 features, including: **Superior vocabulary access** – point-and-click assembly of commands and object specifications; command window for experimentation. **Shortcuts and extended editing capabilities** – extensive drag-and-drop, six-function find-and-replace; navigation markers; script library collection facility; many other timesavers for faster scripting. **Interactive debugging** – comprehensive variable watcher, expression evaluation, enhanced trace log, and real single step debugging! Other features include: integration with FaceSpan and background processing. Unlike other scripting tools, which are either based on the original Script Editor concept, or are designed to look more like traditional programming tools, the designers of Scripter understood from the outset that scripting is different from writing C code. Scripter will change the way you work with AppleScript. Main Event was the first to demonstrate AppleScript tools, including debugging, and is in use by individuals and corporate users. From expert script design to user-friendly editing and implementation, Scripter is the natural companion to AppleScript for all levels of proficiency. $199

TCP/IP Scripting Addition™ is the latest version of an award-winning AppleScript scripting addition (first place in the 1994 "Best Hack" category in the Everyday AppleScript™ Programming Competition). This scripting addition (or osax) allows you to write scripts using MacTCP™ commands in AppleScript™. Potential uses of this include sending e-mail or files through a script, checking if users are logged on (via Finger), automating FTP, Gopher, NetNews, Telnet, and LPR, verifying links in HTML documents, and quickly writing many other TCP/IP client-server programs. Sample scripts are included already implementing many of these functions. When combined with FaceSpan, the potential for rapid implementation of Internet client-server applications is enormous. The TCP/IP Scripting Addition works with AppleScript 1.0 or later and MacTCP 2.0.4 or later. It is compatible with Open Transport™. The TCP/IP Scripting Addition can be used from Script Editor, HyperCard 2.2, MacPERL, FaceSpan and other Open Scripting Architecture applications.

See "http://www.mangotree.com/biz/mango/index.html" for more details. $49

MISCELLANEOUS

NEW! **AppMaker** makes it faster and easier to develop the user interface for a Macintosh application. Just point and click to design your application, then AppMaker creates resources and generates excellent source code. AppMaker supports most development environments including Metrowerks, Symantec, or MPW; C, C++, or Pascal; procedural or object-oriented, using PowerPlant, TCL, or MacApp. The generated code uses the Universal Headers to provide PowerMac compatibility. Beginners use AppMaker to learn object-oriented and Macintosh Toolbox programming techniques. Experts use it to increase productivity. It saves so much time it's like having your own assistant programmer working for you. Includes one-year subscription on CD. $299

BASIC for the Newton is BASIC for the Newton! From NS BASIC Corporation, it is a fully interactive implementation of the BASIC programming language. It runs entirely on the Newton – no host is required. It includes a full set of functions and data types, hand-written input, windows, buttons and extensions to take advantage of the Newton environment. Applications can create files or access the built-in soups. Applications can also access the serial port for input and output. Work directly on the Newton, or through a connected Mac/PC and keyboard. NS BASIC includes a 150 page pocket sized manual. $99

Geekware by Metrowerks is here! In high school, they called you a computer geek. Now, they work at burger joints and wear polyester uniforms. And you don't. Wear it to your favorite burger joint. $24.95

Black Sweatshirts	$29.95
Mousepad	$8.95
Geekware Shirt-Cotton	$24.95
Geekware Shirt-Poly	$24.95
Baseball Cap	$16.95
Inside CodeWarrior 6 Book	$34.95
Blood, Sweat & Code Black Long Sleeve Shirt	$14.95
Cross Platform White Short Sleeve Shirt	$14.95
Arnold Black Long Sleeve Shirt	$14.95

Guide Composer™ gives anyone the ability to create powerful Apple Guide help systems for any new or existing Macintosh application. Great for commercial developers, shareware developers, in-house developers, and consultants, Guide Composer provides a WYSIWYG development environment: Guide content is developed in Guide windows. Design topics, phrases, and panels in the same format as the user will use them. Features are WYSIWYG interface, Topics, phrases, and hierarchical phrases, Coach marks, Fully-Integrated with Apple's Guide Maker (distributed with Guide Composer), compiles scripts automatically, PICTs in Panels, Generated Guide scripts are modifiable, Compiled files are 100% Apple Guide-compatible and royalty-free. Easy-to-use. $99

MachTen UNIX for Macintosh and Power Macintosh MachTen is a Berkeley UNIX that runs on the Classic to the Power Mac, including PowerBooks and Duos! So, in addition to all of the Macintosh applications, you get a Mach-based UNIX with pre-emptive multitasking. MachTen extends the Macintosh operating system with UNIX networking and software development tools. The Macintosh/UNIX integration is so strong that you can even use Macintosh programs and utilities on UNIX data, and UNIX programs and utilities on Macintosh files. Full internet protocol support ensures fast, easy client and server NFS, e-mail, and file transfer between the Macintosh and all TCP-based entities on your network. Built-in internet services include domain name service, POP mail service, internet routing, SLIP & PPP, and Web service. Full X11R5 support with Motif for developing X applications and a high performance X server for using your Mac as an X terminal. MachTen – Power UNIX $695. Personal MachTen (for 68K Macs) $495 . Professional MachTen (for 68K Macs) $695. MachTen X Window Software $350.

For complete product info see the MacTech Web™ site at http://www.mactech.com.

ADVERTISER/ PRODUCT LIST

LIST OF ADVERTISERS

Absoft	23
Adianta Inc.	20
ADInstruments	63
Advantage Learning Systems	64
Aladdin Knowledge Systems Ltd.	5
Aladdin Systems	26
APDA, Apple Computer, Inc.	9
Apple Developer University	62
Ariel Publishing	66
BareBones Software	19
BroadCast Software Corp.	40, 41
Celestin Company	55
Creative Solutions	28
Datapak Software, Inc.	44
Developer Central Macworld	14
dtF Americas, Inc.	47
Eagle Data Protection	22
Emerson Kennedy	62
Engage Communications, Inc.	39
Full Moon Software	63
General Knowledge	59
International Dataware, Inc.	34
Jasik Designs	21
Joint Solutions	16
JYACC	ibc
Language Systems	27
Late Night Software, Ltd.	66
MacTech CD-ROM™, Vol. 1-10	49
MacTech Mail Order Store	45
MacTech Web" Site	53
Macworld Exposition	37
MacXperts	64
Main Event	51
Mango Tree Software, Inc.	50
Mathemaesthetics, Inc.	1
Metrowerks	BC
Micro Macro	11
MindVision Software	32
MultiQuest Corporation	67
Neologic Systems	24, 29
Nisus	42
Nth Degree Software	65
Onyx Technology	55
Quasar Knowledge Systems	IFC
Ray Sauers Associates	52
Scientific Placement	64
Seapine Software, Inc.	67
StepUp Software	54
The SU5 Group, Inc.	48
Symantec	6
Tenon Intersystems	43
The Trattner Network	64
Water's Edge Software	31

LIST OF PRODUCTS

Apprentice • Celestin Company	55
BBEdit 3.1 • BareBones Software	19
BroadCast™ • BroadCast Software Corp.	41
CodeWarrior™ • Metrowerks	BC
DataScript • General Knowledge	59
The Debugger V2 • Jasik Designs	21
Destination: Internet • Celestin Company	55
Developer VISE 3.0 • MindVision Software	32
DragInstall 1 • Ray Sauers Associates	52
dtF, The Relational Database System • dtF Americas, Inc.	47
Duplication Service • International Dataware, Inc.	34
Fortran 77 Software Development Kits • Absoft	23
Fotografika • Celestin Company	55
Installer Pack • StepUp Software	54
JAM® • JYACC	ibc
LS Object Pascal V1.0 • Language Systems	27
MacApp Programmers • MacXperts	64
MACllock • Eagle Data Protection	22
MacForth Plus 4.2 • Creative Solutions	28
MacHASP & Net-MacHASP • Aladdin Knowledge Systems Ltd.	5
MachTen™ • Tenon Intersystems	43
Macintosh Software Development • The SU5 Group, Inc.	48
Mac Modem Cable • Celestin Company	55
MacNosy • Jasik Designs	21
Mac OS SDK • APDA, Apple Computer, Inc.	9
MacRegistry™ • Scientific Placement	64
MENUMILL • Ariel Publishing	66
MicroGuard Plus • Micro Macro	11
neoAccess™ • Neologic Systems	29
NeoShare • Neologic Systems	24
PAIGE™ • DataPak Software, Inc.	44
PatchWorks™ • BroadCast Software Corp.	40
Power MacForth • Creative Solutions	28
PowerTap™ • Emerson Kennedy	62
Programmer Training Courses • Apple Developer University	62
QC: The Macintosh Testing Solution • Onyx Technology	55
QUED/M 2.7 • Nisus	42
Recruitment • Advantage Learning Systems	64
Recruitment • MacXperts	64
Recruitment • Nth Degree Software	64
Recruitment • Scientific Placement	64
Recruitment • The Trattner Network	64
Resorcerer® 1.2 • Mathemaesthetics, Inc.	1
Routers • Engage Communication, Inc.	39
S-CASE™ • MultiQuest Corporation	67
Script Debugger • Late Night Software, Ltd.	66
Scripter • Main Event	51
ScriptGen Pro • StepUp Software	54
ScriptWizard • Full Moon Software	63
SmalltalkAgents® • Quasar Knowledge Systems	IFC
Solutions & Multimedia Developers Guide • Joint Solutions	16
StuffIt InstallerMaker • Aladdin Systems	26
Symantec C++ 8.0 • Symantec	6
TCP/IP Scripting Addition v1.2 • Mango Tree Software, Inc.	50
TestTrack • Seapine Software, Inc.	67
The Macintosh Applications & Parts Developers Guide • Joint Solutions	16
The Memory Mine™ • Adianta Inc.	20
Tool Plus™ 2.6 • Water's Edge Software	31
Trade Show • Developer Central Macworld	14
Trade Show • Macworld Exposition	37
Update Maker 2™ • ADInstruments	63

TIPS & TIDBITS

By Steve Sisak, Contributing Editor

TIP OF THE MONTH

Two Screen MacsBug

One feature of MacsBug that I like is the "swap" command, which lets you leave a second monitor display the debugger (and punches it out of the desktop).

Then, if each DebugStr() ends in ";g" you will get an onscreen log of the last 10 or 15 breaks. I used this to track an elusive bug that would freeze my Mac and trash Macsbug memory... I could see which breakpoint went last.

Matt Slot

Once Bitten Twice Shy

HGetState does not return a valid handle state when you pass it an empty handle (one whose master pointer is NULL). Instead, it returns an error code, so before you call HGetState, be sure to check that the handle isn't empty and execute an alternate code path if it is.

I was bitten by this because I was using HGetState to determine if a handle was to a resource, and the resource had been purged, so HGetState returned an error code instead of the handle flags and I incorrectly thought the handle wasn't to a resource.

This is documented in Inside Macintosh, Memory, p. 1-61, but it bears some repeating. Remeber to check the error values returned by toolbox calls.

Eric Schlegel

Tips & Tidbits continued on page 66

Informant in the Menu Bar

I do drivers, and you just plain can't set a breakpoint in ADB completion routines (freezes the keyboard so MacsBug is worthless!).

So I throw one of the routines below into the routine to see when a piece of code gets executed.

What does it do? It "lights up" a bar (length dependant on screen resolution) in the menu bar. So if you DotToggle(300); you get a flashing short line in the menu bar.

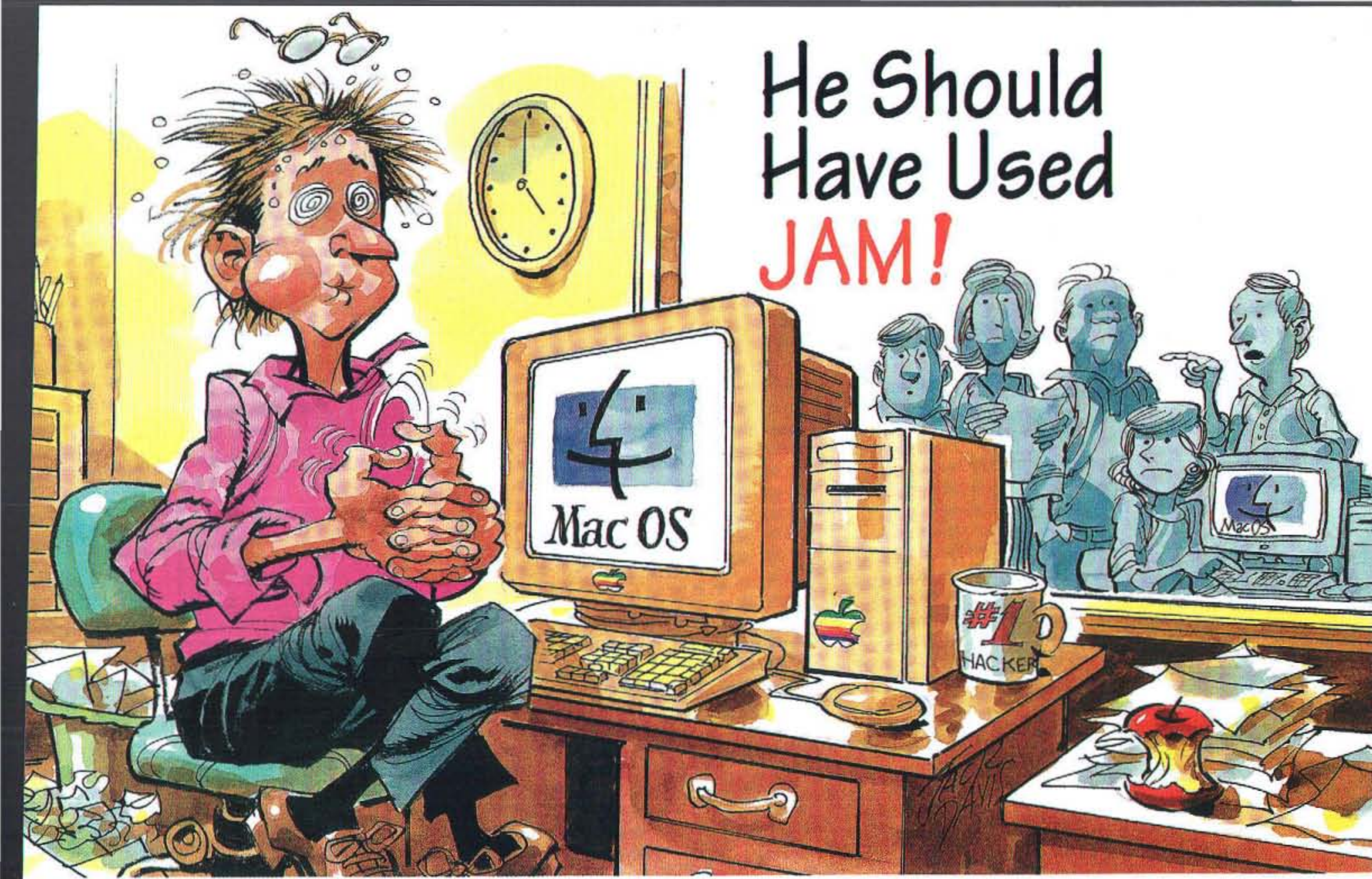
[Leave this out of production code! Beware of using this on 24 bit color screens. Make sure that the value of `where` *will not cause anything to be written into the alpha channel. Your best bet is to test with the screen that's only 8 bits deep. -sgs]*

```
void DotOn(long where) {
   long  *dot;
   dot = (long *)(LMGetScrnBase() + where);
   *dot |= -1;
}
void DotOff(long where) {
   long  *dot;
   dot = (long *)(LMGetScrnBase() + where);
   *dot &= 0;
}
void DotToggle(long where) {
   long  *dot;
   dot = (long *)(LMGetScrnBase() + where);
   *dot ^= -1;
}
```

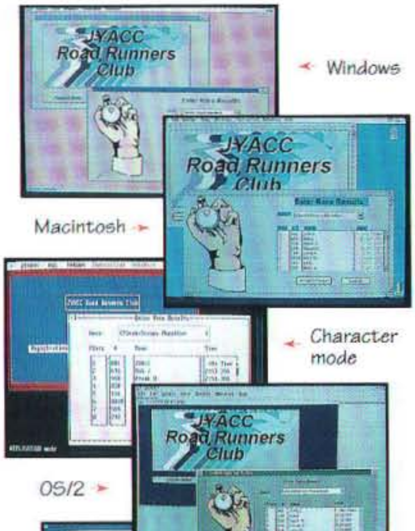
Dave Fleck

*Send us your tips or we'll install EvenBetterBusError on your machine! On the other hand, we might just pay you **$25** for each tip we use, or **$50** for Tip of the Month. You can take your award in goods, subscriptions or US$. Make sure any code compiles, and send tips (and where to mail your winnings) to our **new Tips e-mail address** at tips@mactech.com.*

See page two for our other addresses.



Why Hold Your Breath Waiting for a Client/Server Development Tool? – Get JAM® for the Macintosh Today!



Finally...an industrial strength client/server development tool is available for the Macintosh! JAM allows you to build client/server applications on the Macintosh and deploy them on more than 100 platforms and operating environments including Macintosh, OS/2, Motif, Windows, DOS, VAX/VMS, and virtually every implementation of UNIX. And unlike the competition's runtime-only Mac ports, JAM for Macintosh provides a full application development environment with true Mac look-and-feel. So don't hold your breath any longer - get JAM for the Mac - and get the job done!

JAM gets your Macintosh applications up and running with:

- ✓ Native Mode on Both 68K and Power Macintosh
- ✓ Graphical Screen Editor
- ✓ True Mac Look-and-Feel
- ✓ Full Development Environment -- not just Runtime
- ✓ Portability to OS/2, Motif, Windows, and Character-mode
- ✓ Native DBMS Support
- ✓ Automatic SQL Generation
- ✓ Flexibility to Add 3GL or 4GL Routines
- ✓ No Runtime Charges

Call 1-800-458-3313

or E-mail: macjam@jyacc.com for a free demonstration kit.
For international inquiries call:
1-212-267-7722 or FAX 1-212-608-6753

JAM from JYACC

BRAZIL (55) 11 816 6229 • DENMARK (45) 33 32 55 77 • FINLAND (358) 0 162 966 • FRANCE (33) 1 46 92 45 44 • GERMANY (49) 40 79 70 07 0 • HRVATSKA (385) 1 242 116
INDIA (91) 22 283 1188 • ISRAEL (972) 3 921 8320 • ITALY (39) 2 25 52 65 2 • MEXICO (52) 5 550 4500 • RUSSIA (7) 095 288 1924 • SAUDI ARABIA (9662) 671 8749
SINGAPORE (65) 220 8322 • SLOVENIA (386) 61 1405 004 • SPAIN (34) 1 804 0625 • SWEDEN (46) 8 96 10 10 • SWITZERLAND (41) 21 991 9041 • THAILAND (66) 2 513 3559
THE NETHERLANDS (31) 70 320 9214 • UNITED KINGDOM (44) 171 814 6660

JAM is a registered trademark of JYACC, Inc. Other trademarks are the property of their respective owners.

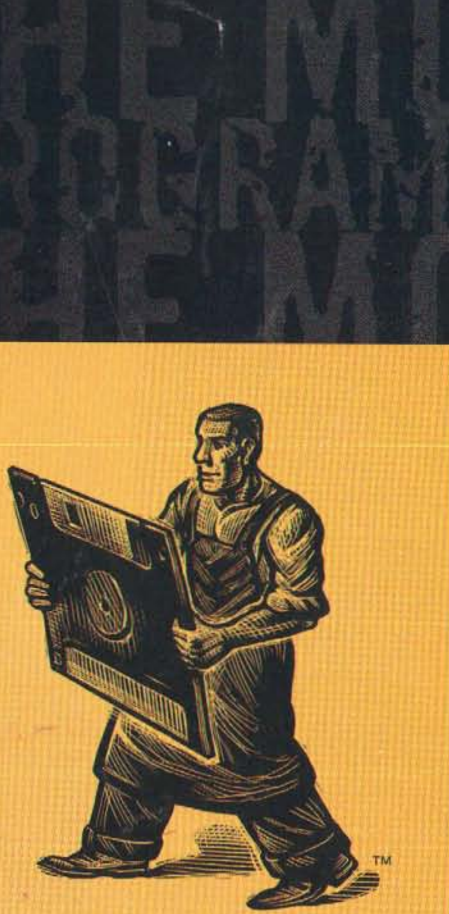
JYACC
Meeting the needs of professional developers for over 16 years

CODEWARRIOR

68K AND POWER MACINTOSH™, MAGIC CAP™, WINDOWS 95™ AND WINDOWS NT™/X86

THE MOST VERSATILE PROGRAMMING ENVIRONMENT IN THE WORLD

GOLD 7 $399 STARTER KIT $79
BRONZE 7 $149 INSIDE POWERPLANT $34.95
INSIDE CODEWARRIOR 7 $34.95



CodeWarrior™

metrowerks®

MACUSER EDITORS' CHOICE AWARD · WINNER · 1994

1994 SOFTWARE
PRODUCT
OF THE YEAR

To order contact Metrowerks:
voice: (512) 305-0400 fax: (512) 305-0440 e-mail: sales@metrowerks.com
World Wide Web: http://www.metrowerks.com

Metrowerks, the Metrowerks logo, PowerPlant and CodeWarrior are registered trademarks of Metrowerks Inc.
All other companies and products are trademarks of their respective holdings and are hereby recognized.